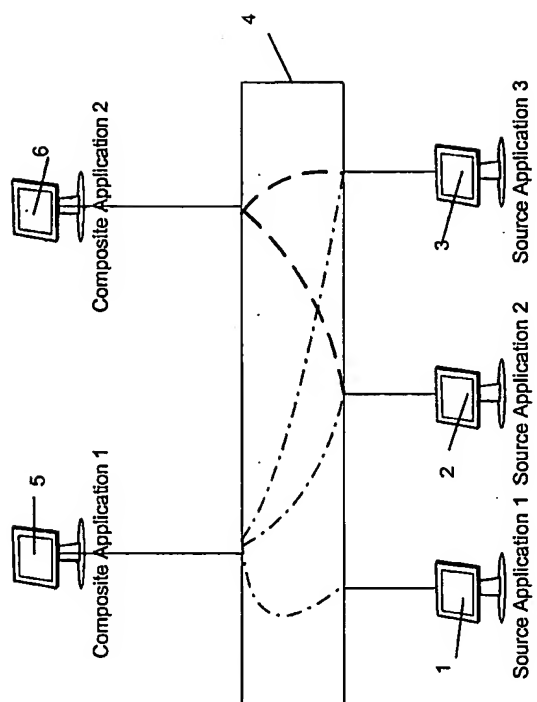
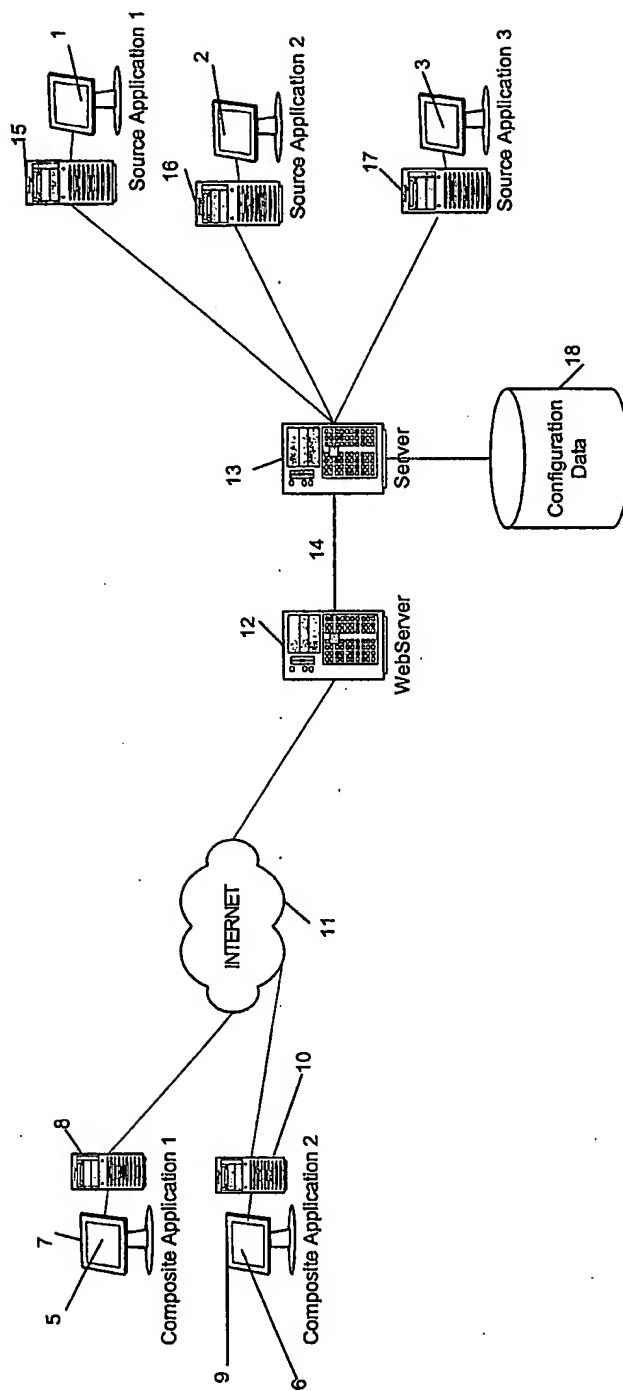


1/48



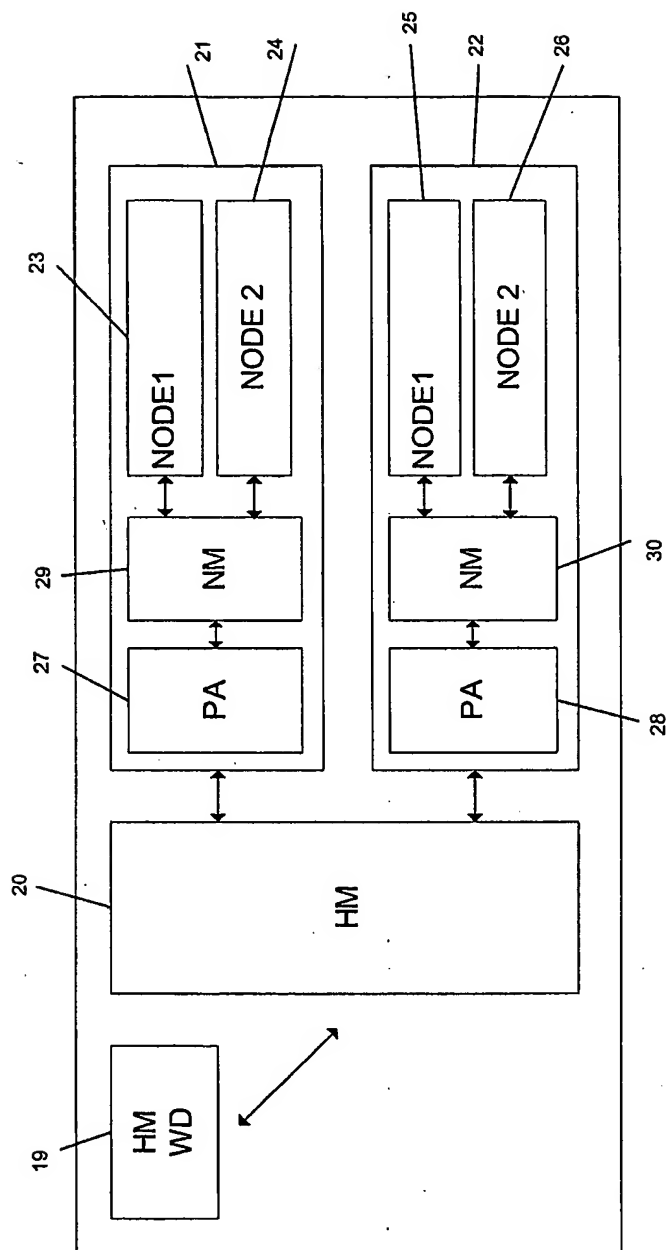
**FIG 1**

2/48

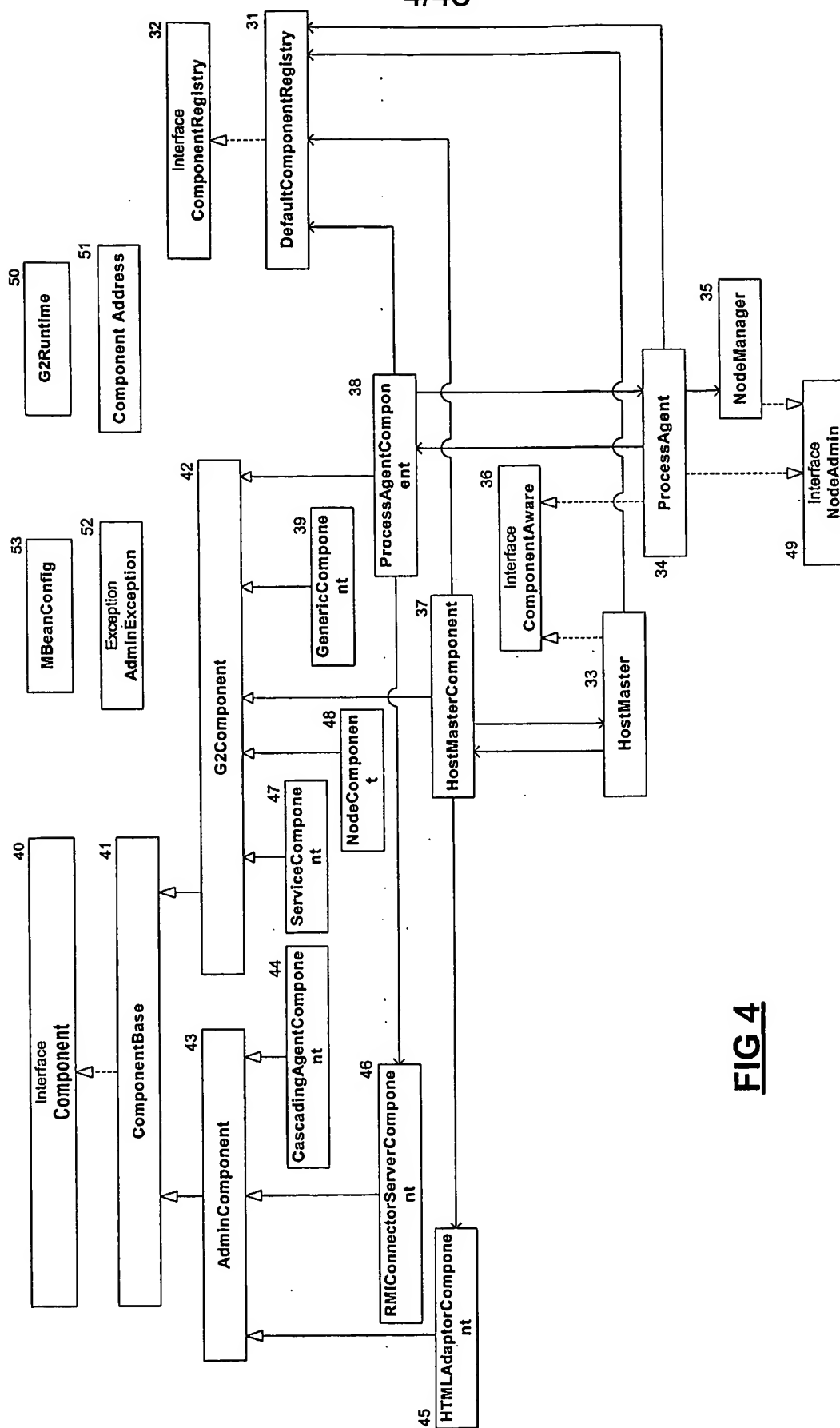


**FIG 2**

3/48



**FIG 3**



**FIG 4**

5/48

31

DefaultComponentRegistry
-DefaultComponentRegistry minstance=new DefaultComponentRegistry() -MbeanServer mMyMBeanServer -String mG2MbeanDomain -String mDefaultDomain -String mHostIpString -String mProcessIdString -Hashtable mRegisteredComponent
-DefaultComponentRegistry() -void Init() -void registerMBeanObject( String domain, Object obj, Hashtable nvps) -void unregisterMBeanObject(ObjectName target) -void CleanUpMBeanServer() -void registerComponent(Component c) -void unregisterComponent(Component c) -Component registerComponentAware(ComponentAware ca) -Component registerGenericComponent(Object o) -Component registerRmiConnectorServer(RmiConnectorServer o) -Component registerCascadingAgent(CascadingAgent o) -Component registerHtmlAdapterServer(HtmlAdapterServer o) -void cleanup() -Set findObjectNames(AddressMask am) +Hashtable getComponentProperties(Component c) +ComponentRegistry getInstance() +ComponentAddress lookupComponentAddress(Object o) +void waitForComponentRegistration(String objectname, int maxwaitseconds) +Component registerObject(Object o) +void unregisterObject(Object o) +void addNotificationListener(ObjectName name, NotificationListener listener, java.lang.Object handback) +void InvokeOnObject(AddressMask m, String method, Object[] params, String[] signature) +void InvokeOnObject(Object o, String method, Object[] params, String[] signature) +void InvokeOnObject(ComponentAddress addr, String method, Object[] params, String[] signature) +Set findLocalObjectComponents(AddressMask props) +Set findGlobalObjectComponents(AddressMask props)

**FIG 4A**

32

ComponentRegistry
+Component registerObject(Object c) +void unregisterObject(Object c) +void InvokeOnObject(AddressMask addr, String method, Object[] params, String[] signature) +void InvokeOnObject(ComponentAddress addr, String method, Object[] params, String[] signature) +void InvokeOnObject(Object o, String method, Object[] params, String[] signature) +void WaitForComponentRegistration(String objectname, int maxwaitseconds) +ComponentAddress lookupComponentAddress(Object o) +Set findLocalObjectComponents(AddressMask props) +set findGlobalObjectComponents(AddressMask props)

**FIG 4B**

6/48

33

HostMaster
-boolean misDebug -int mNextProcessId -Vector mActiveProcessIds -boolean misAlive -ComponentRegistry mCRegistry -HostMaster mTheMaster -HostMasterComponent mHotMasterComponent -String mProcessPropsString
+HostMaster() +int addProcess() +void destroyProcess(Integer pid) +void main(String[] args) +void shutdown() +void kill() +void SetIsDebug(boolean d) +boolean isDebug() +boolean isAlive() -boolean getBooleanArg(String token, String[] args) +void processLost(int pid) +Class getComponentClass() -void recover() -ShutdownMonitor

**FIG 4C**

34

ProcessAgent
-int mMyPid -ProcessAgentComponent mAgent -ProcessAgent mAgent -NodeManager mNodeManager -boolean misAlive -ComponentRegistry mCRegistry
+ProcessAgent(int pid) +boolean isAlive() +void main(String[] args) +void addNode(NodeType nt) +void removeNode(NodeType nt) +void startNode(NodeType nt) +void stopNode(NodeType nt) +void destroyProcess() +void addNode(String nt) +void removeNode(String nt) +void startNode(String nt) +void stopNode(String nt) +int getProcessId() +Class getComponentClass()

**FIG 4D**

7/48

35

NodeManager
-Vector mynodes
+void NodeManager() +Node createNode(NodeType type) -void destroyNode(Node n) -void startNode(Node n) -void stopNode(Node n) +void destroyNode(NodeType nt) +void startNode(NodeType nt) +void stopNode(NodeType nt) +Node getNode(NodeType type) +Integer activeNodeCount() +void recoverNodes()

**FIG 4E**

36

ComponentAware
+Class getComponentClass()

**FIG 4F**

37

HostMasterComponent
-String MBEANTYPE="HostMaster" -String MYBEAQNDDESCRIPTION -int HEARTBEATRETRIES=1 -int HEARTBEATPERIOD=5000 -HostMaster mHostMaster -HtmlAdapterComponent mHtmlAdapterComponent -Hashtable mRemAgents -ComponentRegistry mCRegistry
+HostMasterComponent(Object hm) -CascadingAgent createCascadingAgent(String cascadingDomain, String hostname, int targetProcess) +String getType() +void processAdded(int pid) +void processRemoved(int pid) +void preRegister() +void preUnregister() +void handleNotification(Notification n, Object o) +void processRecovered(int pid) #String getMBeanDescription() #ModelMBeanOperationInfo[] getMBeanOperations()

**FIG 4G**

8/48

38

ProcessAgentComponent
+String MBEANTYPE="ProcessAgent" -String MYBEANDESCRIPTION="" -ProcessAgent mProcessAgent -RMICConnectorServerComponent mRMIServer -ComponentRegistry mCRegistry
+ProcessAgentRegistry(Object pa) + void preRegister() +void preUnregister() +String getType() #String getMBeanDescription() #ModelMBeanOperationInfo[] getMBeanOperations()

**FIG 4H**

39

GenericComponent
-String MYBEANDESCRIPTION="Object" +GenericComponent(Object o) +String getType() #String getMBeanDescription()

**FIG 4I**

40

Component
+String IDPROP="id" +String TYPEPROP="type" +String HOSTPROP="host" +String PROCESSPROP="process" +String NODEPROP="node" +String SERVICEPROP="service"
+String getId() +String getType() +String getService() +String getNode() +String getAdminDomain() +Object getObject() +Object getMBean() +String getHostId() +String getProcessId() +void preRegister() +void preUnregister()

**FIG 4J**



9/48

41

ComponentBase
-Object mTheObject -String mObjectDescription -Object mTheMBean -ArrayList mMBeanOperationInfo -ArrayList mMBeanAttributeInfo -ArrayList mMBeanNotificationInfo -String mHostIp -String mProcessId
+ComponentBase(Object o) +boolean isMBean(Object o) +String getId() +String getType() +String getService() +String getNode() +Object getObject() +void preRegister() +void preUnregister() +Object getMBean() #void setObjectDescription(String desc) #ModelMBeanOperatingInfo[] getMBeanOperations() #ModelMBeanAttributeInfo[] getMBeanAttributes() #ModelMBeanNotificationInfo[] getMBeanNotifications() #String getMBeanDescription() -void setupMBeanInfo() +String getAdminDomain() +String getHostIP() +String getProcessId()

**FIG 4K**

42

G2Component
+G2Component(Object o) +String getAdminDomain()

**FIG 4L**

43

AdminComponent
+AdminComponent(Object o) +String getAdminDomain()

**FIG 4M**

10/48

44

CascadingAgentComponent
+String MBEANTYPE="CascadingAgent"
+String CONNECTORCLIENTCLASSNAME="com.sun.i"
+CascadingAgentComponent(CascadingAgent agent)
+String getId()
+String getType()

**FIG 4N**

45

HTMLAdaptorComponent
-String HTTPCONNECTIONSERVER
-HtmlAdaptorServer mHtmlAdaptor
+HtmlAdaptorComponent(HtmlAdaptorServer has)
+String getType()

**FIG 4O**

46

RMIconnectorServerComponent
+String RMICONNECTORSERVER="RmiConnectorServer"
+String RMICONNECTORSERVICENAMEPREFIX="Process-"
-RMIconnectorServer mTheRmiConnector=null
+RMIconnectorServerComponent(RmiConnectorServer o)
+String getRMIconnectorServiceName(int targetProcessId)
+String getType()

**FIG 4P**

47

.ServiceComponent
-BaseService m_Service
-String MYBEANDESCRIPTION="A Service"
+ServiceComponent(Object s)
+String etType()
+String getNode()
#String getMBeanDescription()
#ModelMBeanAttributeInfo[] getMBeanAttributes()
#ModelMBeanOperationInfo[] getMBeanOperation

**FIG 4Q**

11/48

48

NodeComponent
-String MYMBEANDESCRIPTION="A node : runs a set" -MasterNode mNode
+NodeComponent(Object n) +String getType() #String getMBeanDescription() #ModelMBeanAttributeInfo[] getMBeanAttributes() #ModelMBeanOperationInfo[] getMBeanOperations()

**FIG 4R**

49

NodeAdmin
+Node createNode(NodeType type) +void startNode(NodeType nt) +void stopNode(NodeType nt) +Node getNode(NodeType type) +Integer activeNodeCount() +void recoverNodes()

**FIG 4S**

50

G2Runtime
-G2Runtime mInstance=new G2Runtime() -Int mProcessId -String mHostIdString
-G2Runtime() +void init() +ComponentRegistry getComponentRegistry() +String getHostId() +int getProcessId()

**FIG 4T**

12/48

51

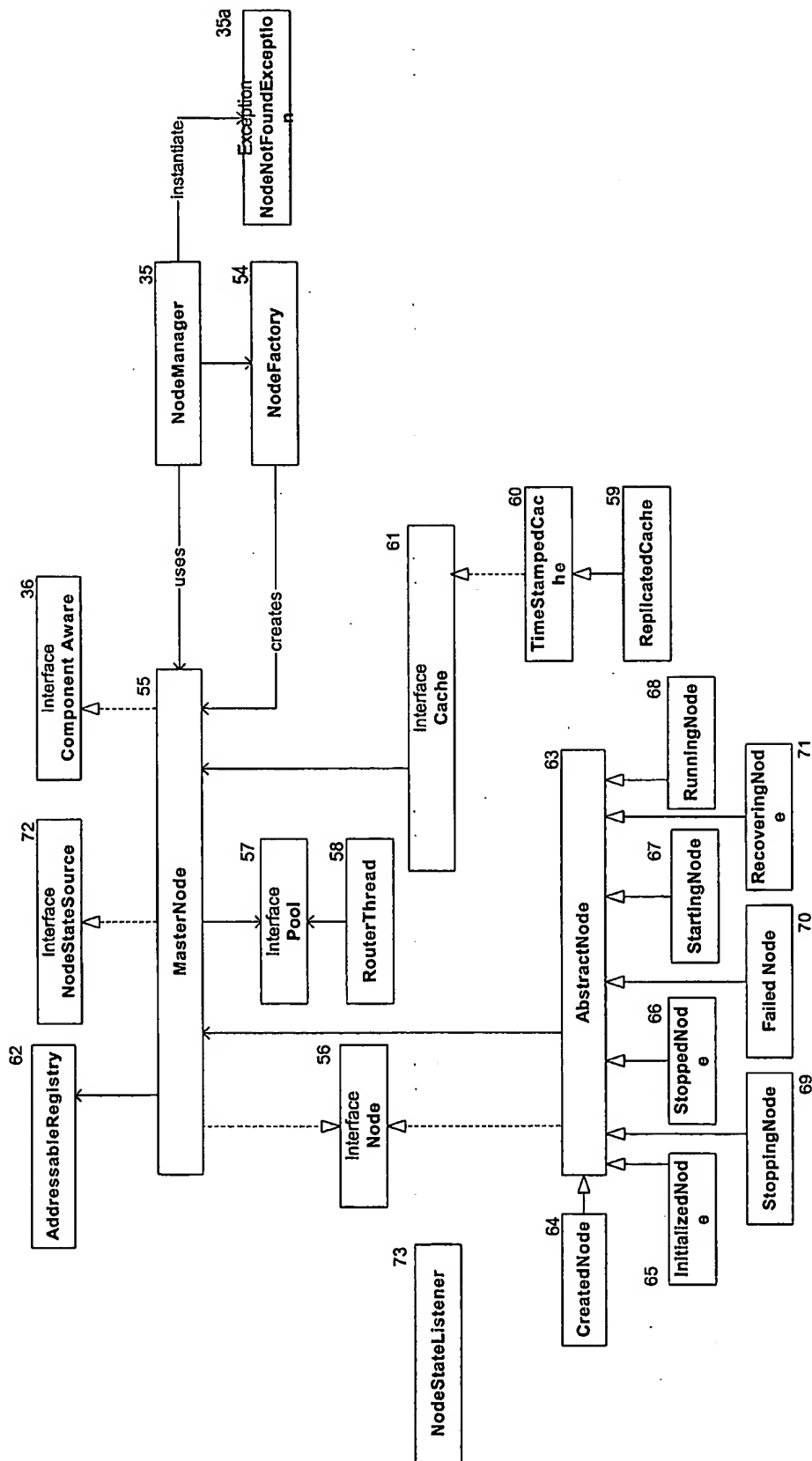
ComponentAddress
-Hashtable mAddressAttributes -String HOST=G2Runtime.getHostIp() -String PROCESS="" + G2Runtime.getProcessId() -AddressAttribute[] mAttributeList=new AddressAttribute[]
+ComponentAddress newAddress() +ComponentAddress getAddress(Component c) -ComponentAddress(Hashtable props) -ComponentAddress() +Hashtable getAttributes() +void setAttribute(AddressAttribute aa) +AddressAttribute getAttribute(String name) +boolean isComplete() +boolean equals(Object o) +String toString()

**FIG 4U**

52

MBeanConfig
+int RMIPORT=1099 +int HTTPPORT=8080 +String G2DOMAIN="G2Domain" +String DEFAULTDOMAIN="DefaultDomain" +String PIDPROPERTYNAME="g2.processId"

**FIG 4V**



**FIG 5**

14/48

54

```

com.genient.platform.node.NodeFactory
-Object[] NODEDEFS={ { NodeType.A
-NodeFactory instance=new NodeFactory()
# NodeFactory()
+NodeFactory getInstance()
+Node[] createNodes(NodeType[] nodetype)
+Node createNode(NodeType nodatype)
-void loadNodeDefs()
-LinkedList getApplicationAdapterServiceList()
-LinkedList getClientAdapterServiceList()
-LinkedList getBPStoreServiceList()
-LinkedList getTMSServiceList()
-LinkedList getSystemManagerServiceList()
-LinkedList getDebugNodeServiceList()

```

**FIG 5A**

56

```

Interface
...platform.node.Node
+void init()
+void start()
+void stop()
+void destroy()
+void recover()

```

**FIG 5C**

57

```

Interface
com.genient.platform.util.Pool
+void releaseObject(Object o)
+Object getObject()
+int getAvailableCount()
+int getSize()

```

**FIG 5D**

55

```

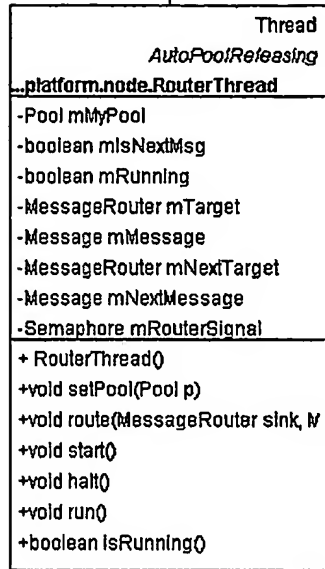
ComponentAware
...platform.node.MasterNode
-Thread mEventNotifier
-NodeType type
-String subType
-HashMap mStateTransitions
-HashMap mServices
-HashMap mListeners
-boolean isInitialised
-boolean removed
-String lastError=NOERROR
-String NOERROR="NONE"
-Node mState
-Pool mRouterThreadPool
-Cache mCache
-AddressableRegistry mAddressat
-void notifyListeners()
-void setType(NodeType type)
-void startServices()
-void stopServices()
+String getLastError()
+NodeType getType()
+Node getState()
+String getStateDescription()
-void initRouterThreadPool()
#void addService(HandlerContext s
#void removeServices()
+ MasterNode()
+void addListener()
+void removeListener()
+void start()
+void stop()
+void setSubType(String st)
+String getSubType()
+void init()
+void destroy()
+Integer getAvailableRouterThread
+RouterThread getRouterThread()
+void releaseRouterThread(Router
+Class getComponentClass()
+void recover()
#void setState(Node newState)

```

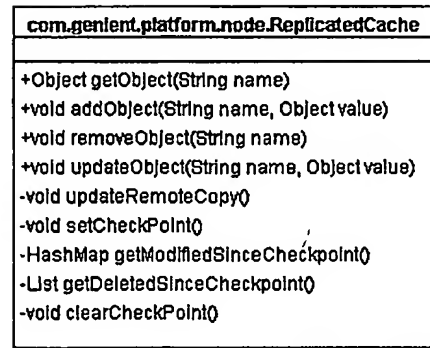
**FIG 5B**

15/48

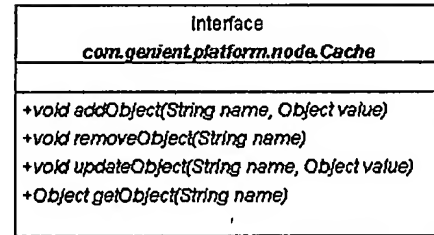
58

**FIG 5E**

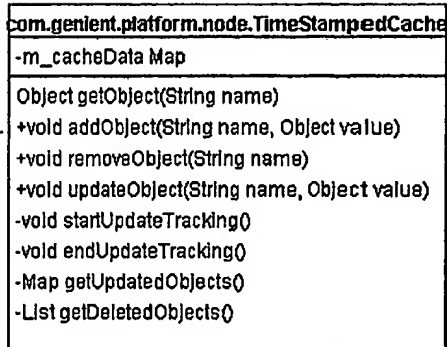
61

**FIG 5F**

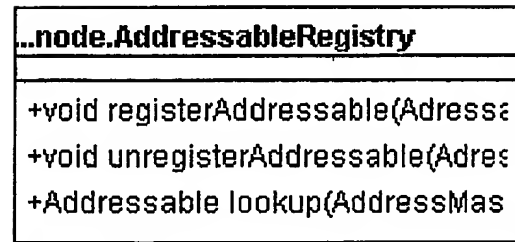
62

**FIG 5H**

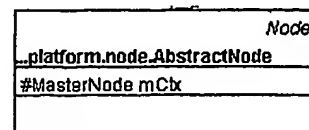
60

**FIG 5G**

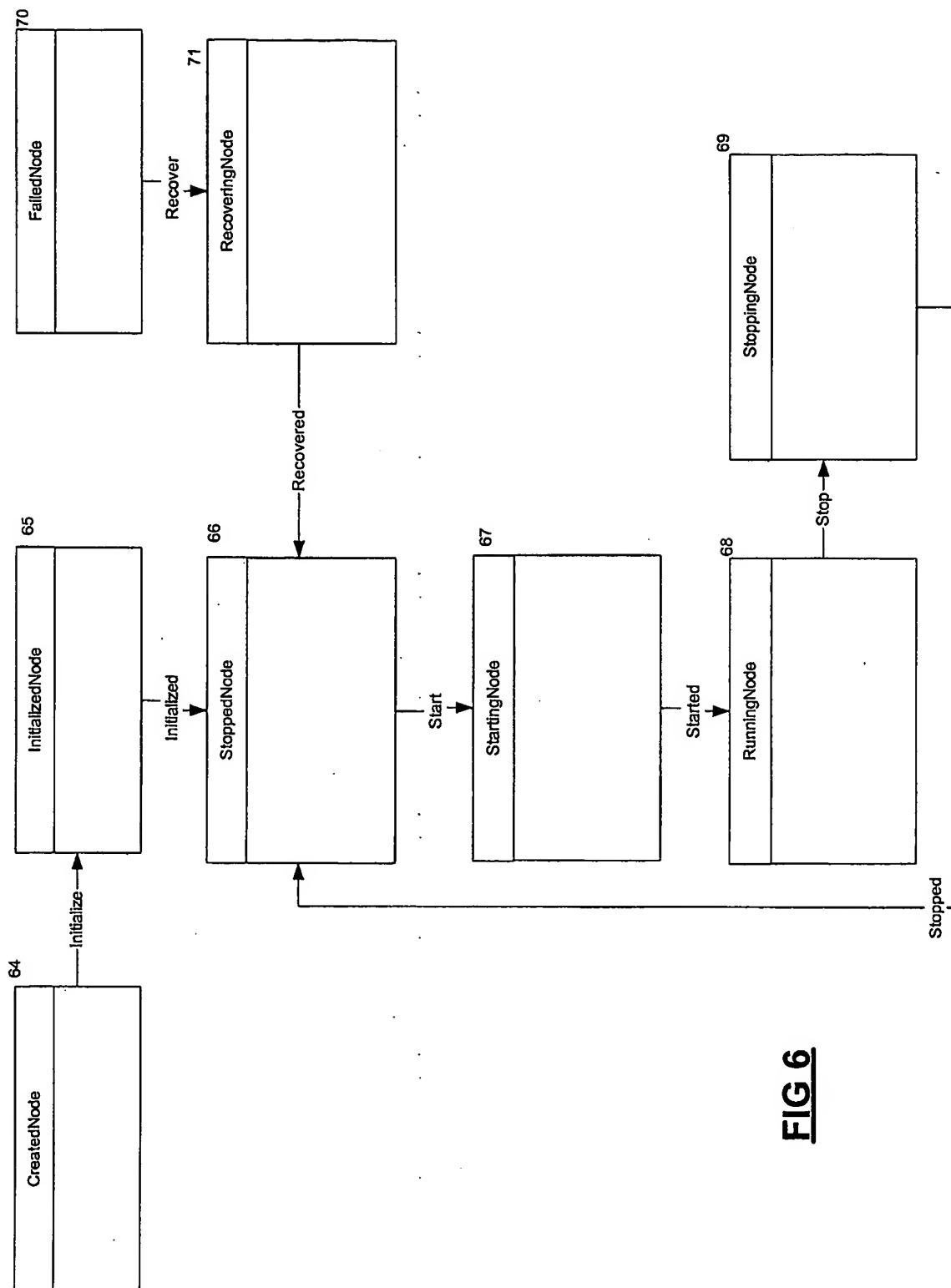
63

**FIG 5I**

77

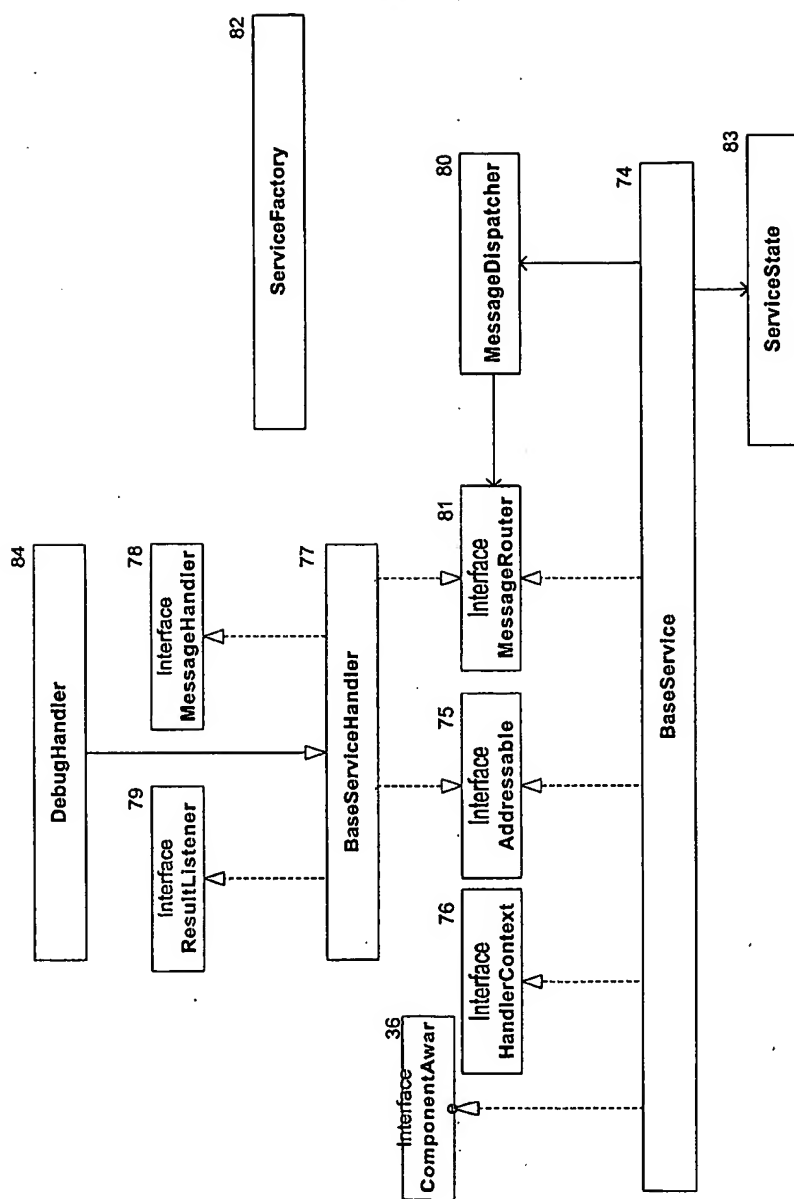
**FIG 5J**

16/48



**FIG 6**





**FIG 7**

<i>ComponentAware</i>	74
<i>...platform.framework.BaseService</i>	
<ul style="list-style-type: none"> <li>-Int mMessageSendCount</li> <li>-int mMessageReceivedCount</li> <li>-float mMsgAvg</li> <li>-HashMap mMessageListeners</li> <li>-int mResultsHandled</li> <li>-Int mResultsMissed</li> <li>-MasterNode mNode</li> <li>-ServiceType mType</li> <li>-ServiceState mState</li> <li>-Pool mServiceableHandlerPool</li> <li>-PriorityQueue mInQueue</li> <li>-PriorityQueue mOutQueue</li> <li>-MessageDispatcher mMyInDispat</li> <li>-MessageDispatcher mMyOutDispa</li> <li>-ComponentAddress mMyAddress</li> <li>-AddressableRegistry mRegistry</li> </ul>	
<<constructor>>	
<ul style="list-style-type: none"> <li>+ BaseService(ServiceType service</li> <li>+void initService(String host, String</li> <li>+ServiceType getType()</li> <li>#void sendServiceMessage(Messa</li> <li>+void route(Message msg)</li> <li>+void updateMessageStatistics(Me</li> <li>+ComponentAddress getAddress()</li> <li>#void initHandlerPool(Vector handl</li> <li>+void startService()</li> <li>+void stopService()</li> <li>+MasterNode getNode()</li> <li>+Integer getState()</li> <li>+Integer getInQueueSize()</li> <li>+Integer getOutQueueSize()</li> <li>+Integer getMessageInCount()</li> <li>+Integer getMessageOutCount()</li> <li>+String getAverageMessageAge()</li> <li>+Integer getAvailableHandlers()</li> <li>#Queue getInQueue()</li> <li>#Queue getOutQueue()</li> <li>+void addResultListener(MessageId</li> <li>+ResultListener removeResultListe</li> <li>+Integer getResultListenersCount()</li> <li>+void receivedResult()</li> <li>+Integer getResultsHandled()</li> <li>+void missedResult()</li> <li>+Integer getResultsMissed()</li> <li>+Class getComponentClass()</li> </ul>	
-MessageToHandlerForwarder	
-MessageToRouterForwarder	

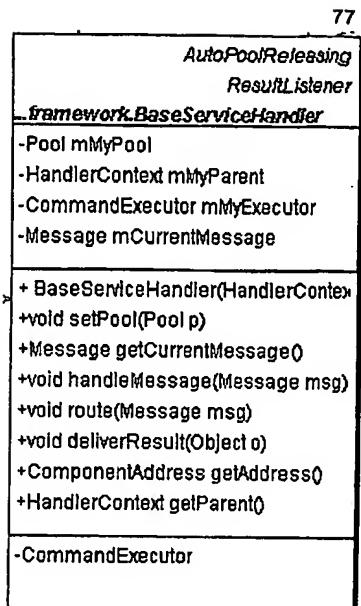
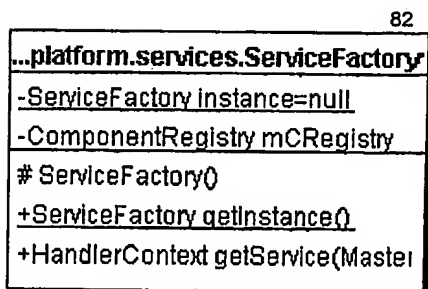
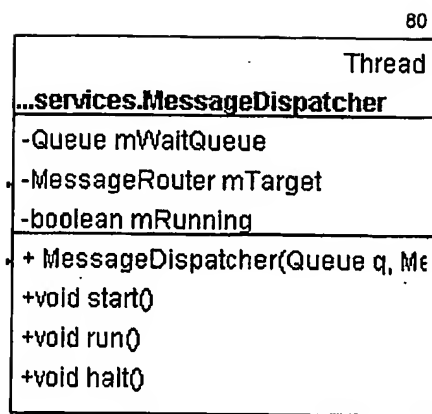
**FIG 7A**

18/48

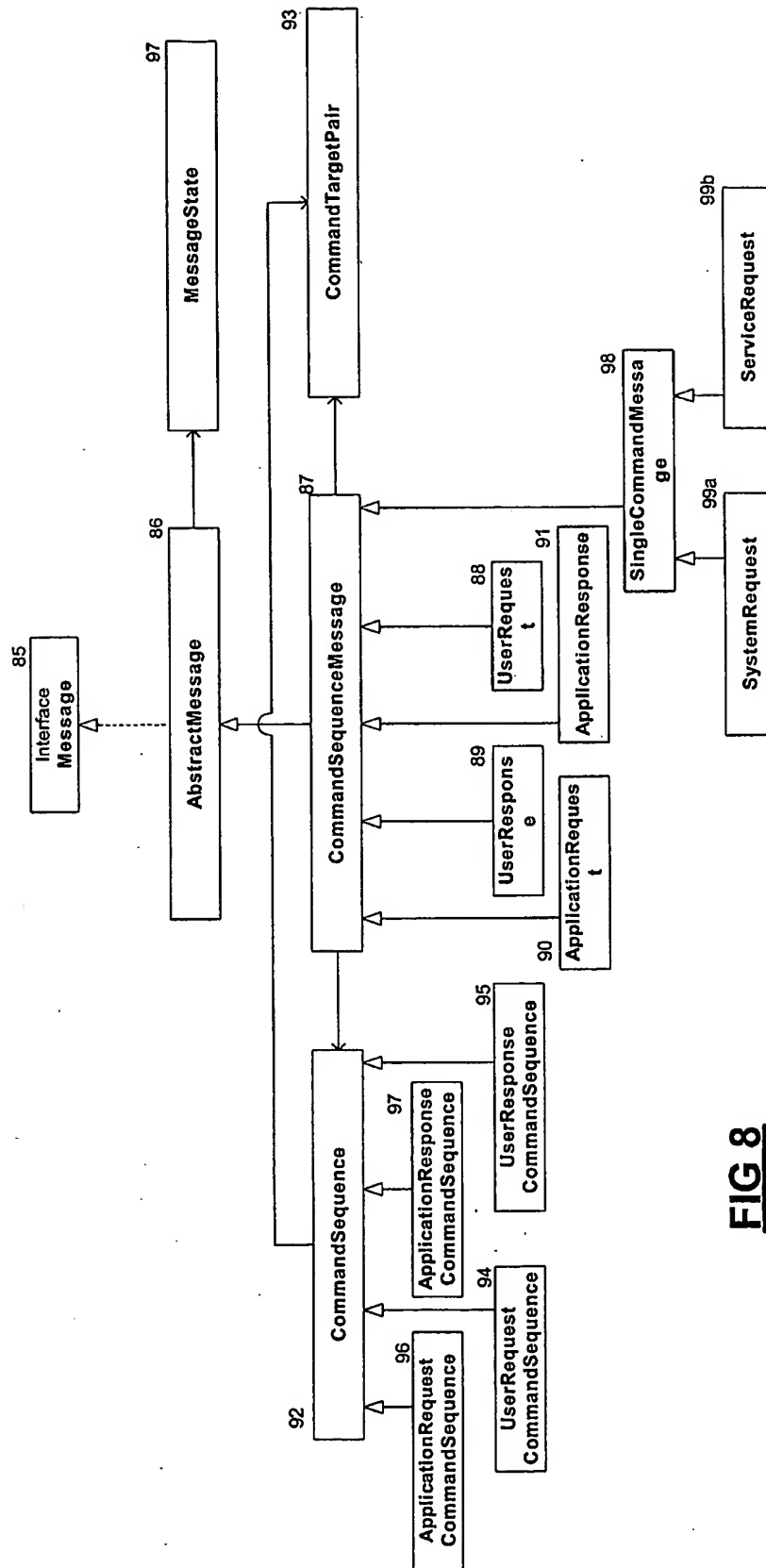
Interface	76
<i>com.genient.platform.framework.HandlerContext</i>	
<ul style="list-style-type: none"> <li>+ServiceType getType()</li> <li>+void route(Message msg)</li> <li>+ComponentAddress getAddress()</li> <li>+Integer getState()</li> <li>+void addResultListener(MessageId replyToId, ResultListener target)</li> <li>+ResultListener removeResultListener(MessageId replyToId)</li> <li>+void registerAddressable(Addressable obj)</li> <li>+void unregisterAddressable(Addressable obj)</li> <li>+ArrayList lookupAddressables(AddressMask mask)</li> <li>+void addResultListener(MessageId replyToId, ResultListener target)</li> <li>+void addPersistentResultListener(MessageId replyToId, ResultListe</li> </ul>	

**FIG 7B**

19/48

**FIG 7C****FIG 7E****FIG 7D****FIG 7F**

20/48



**FIG 8**

AbstractMessage
<pre> -mClonedFrom : MessageId -mCreateTime : long -mHeaderMap : HashMap -mHeaders : HashMap -mHistory : ArrayList -mId : MessageId -mNVPs : HashMap -mSource : AddressMask -mState : MessageState  #AbstractMessage(src:Addressable, cb:GenIntId) +void addMessageNVP(name:String, value:Serializable) +void appendToHistory (id: MessageId) +void cancel() +void commandFailed (e:RequestProcessingException) +boolean commandFinished () +void commandSucceeded() +void commandSuccessfulNotification() +Message copyWithNewId() +AddressMask getAddressMask() +long getAge() +Command getCommand() +Command getCommandForState (state:MessageState) +GenIntId getContext() +Object getHeaderValue (headername : String) +Iterator getHistory() +MessageId getId() +Serializable getMessageNVPValue (name : String) +MessageId getRootId() +AddressMask getSource() +AddressMask getTargetForState(state: MessageState) +boolean hasFinished() +Map headersAsMap() +void initHeaderMap() +boolean isMultiDelivery() +boolean isValidHeaderName (name: String) +boolean isValidHeaderType (name: String, value: Object) +void leaveProcess() +void readObject (stream: ObjectInputStream) +void removeMessageNVP (name: String) +void setHeaderValue (headername: String, headervalue:Object) +void setHistory (hist: Iterator) +String toString() </pre>

**FIG 8B**

```

Interface
Message

+void addMessageNVP(name:String, value:Serializable)
+void cancel()
+void commandFailed (e:RequestProcessingException)
+void commandSucceeded()
+Message copyWithNewId()
+AddressMask getAddressMask()
+long getAge()
+Command getCommand()
+GentletID getContext()
+Object getHeaderValue (headername : String)
+Iterator getHistory()
+MessageId getId()
+Serializable getMessageNVPValue (name : String)
+MessageId getRootId()
+MessageType getTypeId()
+boolean hasFinished()
+Map headersAsMap()
+boolean isMultiDelivery()
+boolean isValidHeaderName (name: String)
+void leaveProcess()
+void readObject (stream: ObjectInputStream)
+void removeMessageNVP (name: String)
+void setHeaderValue (headername: String, headervalue:Object

```

**FIG 8A**

87

CommandSequenceMessage
<div>-mSequence : CommandSequence</div> <div>-m_ErrorDelivery : CommandTargetPair</div> <div>-m_IPCDelivery : CommandTargetPair</div> <div>-m_ProcessDelivery : CommandTargetPair</div> <div>+CommandSequenceMessage (src:Addressable, defaultSeq : CommandSequence, errorDelivery : CommandTargetPair, ctx : GenentId)</div> <div>+CommandSequenceMessage (src:Addressable, defaultSeq : CommandSequence, errorDelivery : CommandTargetPair, ctx : GenentId, scrollsource: boolean)</div> <div>#boolean commandsFinished()</div> <div>#void commandSuccessNotification()</div> <div>+Command getCommandForState (state : MessageState)</div> <div>#AddressMask getTargetForState (state: Message State)</div>

FIG 8C

88

UserRequest
<div>+UserRequest (src:Addressable, requestData : Object, ctx : GenentId)</div> <div>+MessageType getType()</div> <div>-CommandTargetPair getFailureDelivery()</div>

FIG 8D

89

UserResponse
<div>+UserResponse (src:Addressable, requestData : Object, ctx : GenentId)</div> <div>+MessageType getType()</div> <div>-CommandTargetPair getFailureDelivery()</div>

FIG 8E

90

ApplicationRequest
<div>+ApplicationRequest (src:Addressable, requestData : Object, ctx : GenentId)</div> <div>+MessageType getType()</div> <div>-CommandTargetPair getFailureDelivery()</div>

FIG 8F

91

ApplicationResponse
<div>+ApplicationResponse (src:Addressable, requestData : Object, ctx : GenentId)</div> <div>+MessageType getType()</div> <div>-CommandTargetPair getFailureDelivery()</div>

FIG 8G

93

CommandTargetPair
-mCommand : Command
-mCommandClass : Class
-mTarget : Address Mask
+CommandTargetPair(cmd : Class, trg : AddressMask)
+Command getCommand()
+AddressMask getTarget()

**FIG 8I**

92

CommandSequence
-mCommandTargetPairs : ArrayList
-mIndex: int
+CommandSequence (ctp : CommandTargetPair[])
+void cancel()
+boolean isFinished()
+CommandTargetPair nextCommandTarget()
+AddressMask peekNextAddress()
+void reset()

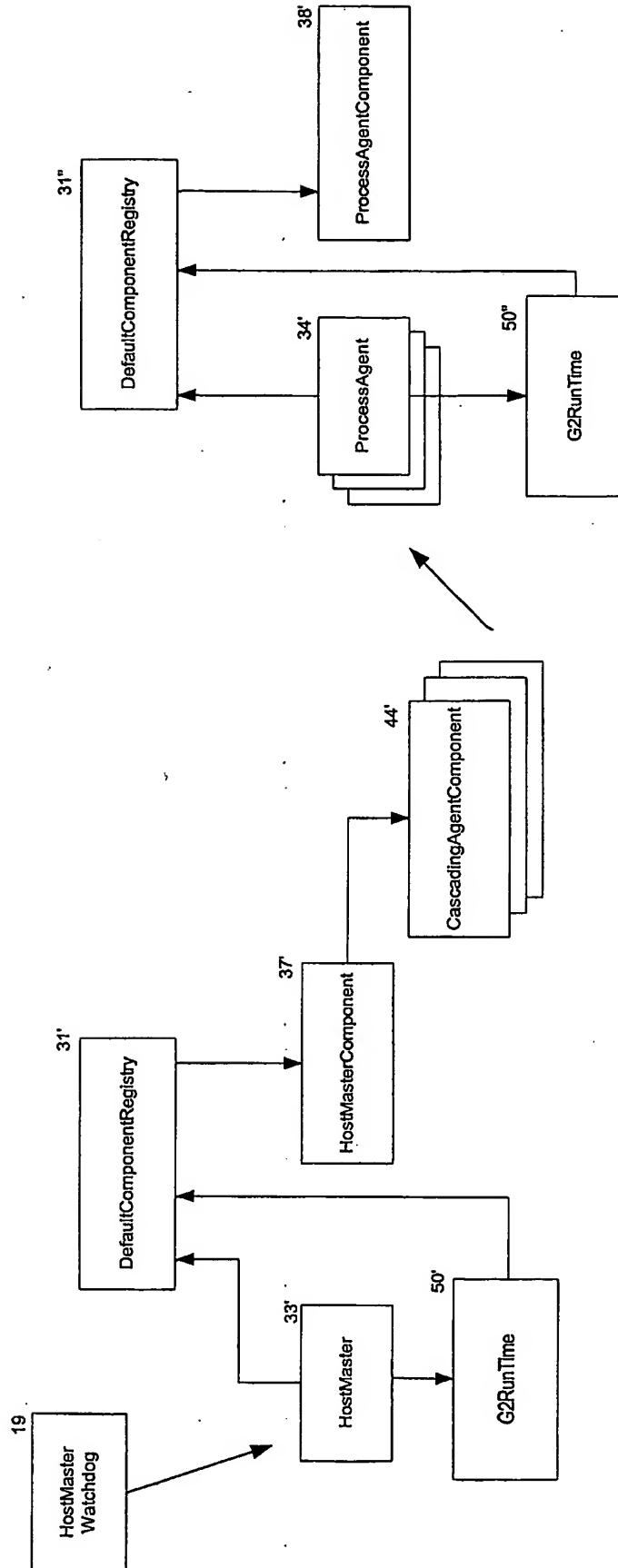
**FIG 8H**

98

SingleCommandMessage
+SingleCommandMessage (src:Addressable, target : AddressMask, cmd: Command, ctx : GenentId, internaliseResponse : boolean)
+SingleCommandMessage (src:Addressable, target : AddressMask, cmd: Command, ctx : GenentId)
-CommandTargetPair getFailureDelivery()
-CommandSequence createCommandSequence (src:Addressable, target : AddressMask, cmd: Command, ctx : GenentId, internaliseResponse : boolean)

**FIG 8K**

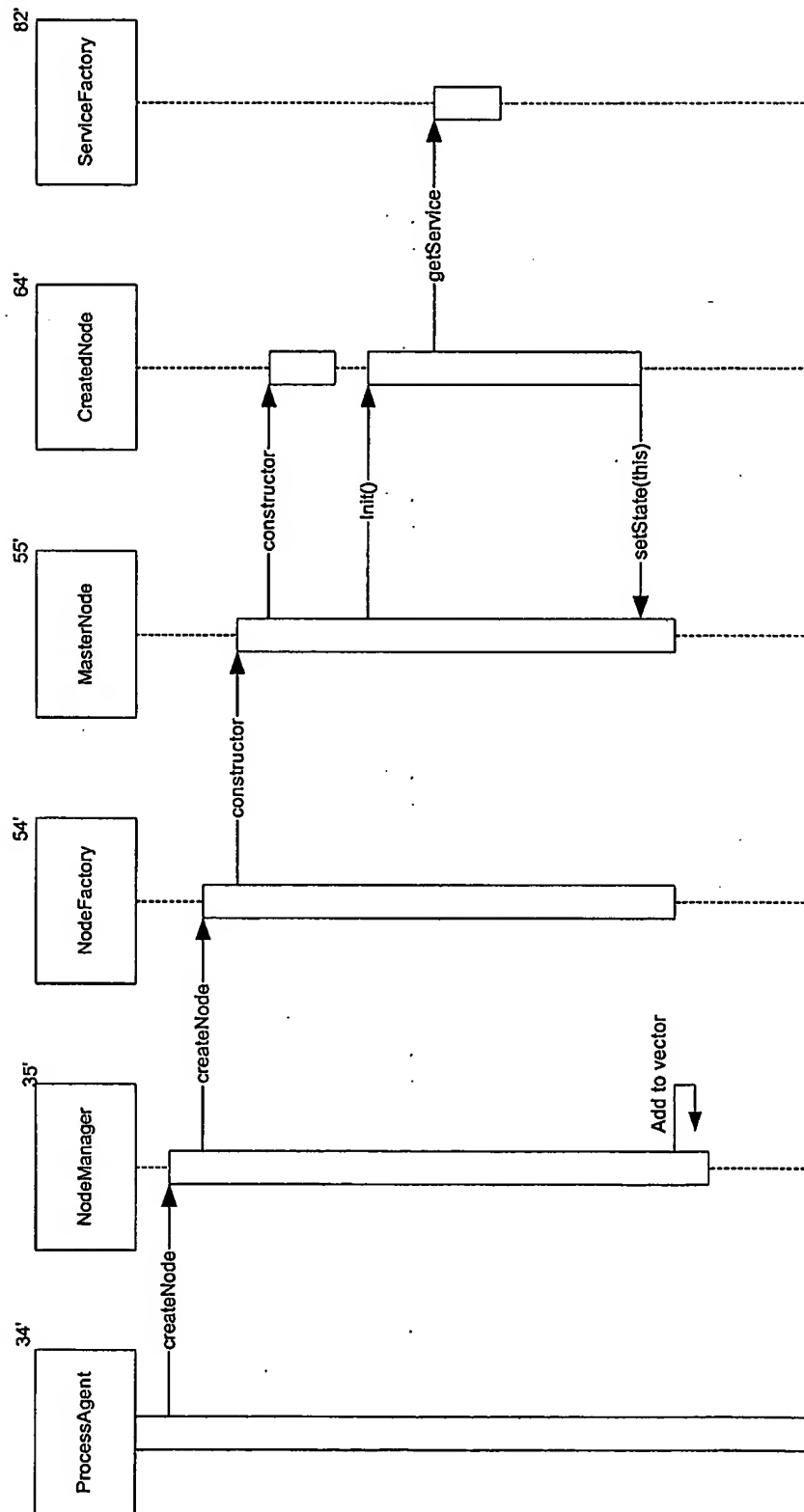
24/48



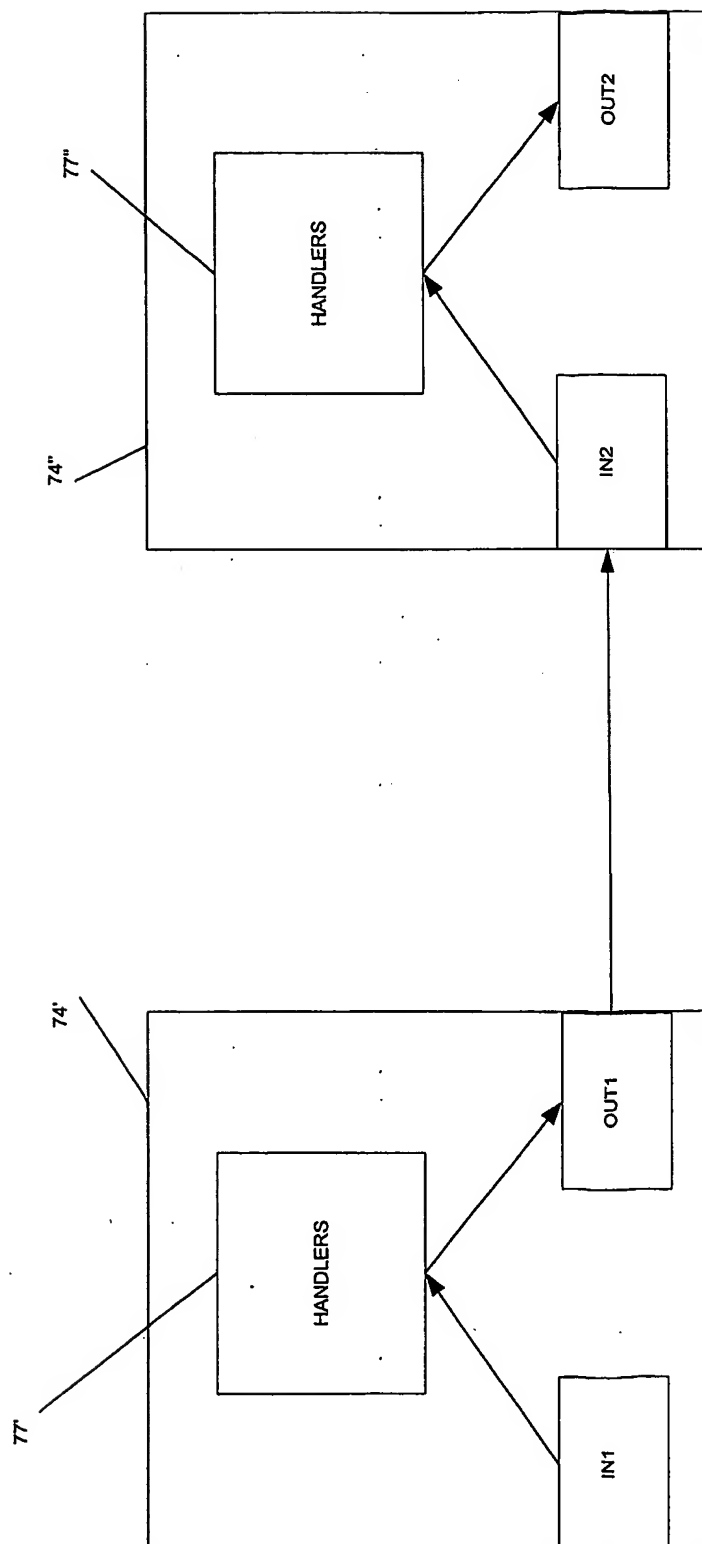
**FIG 9**



25/48

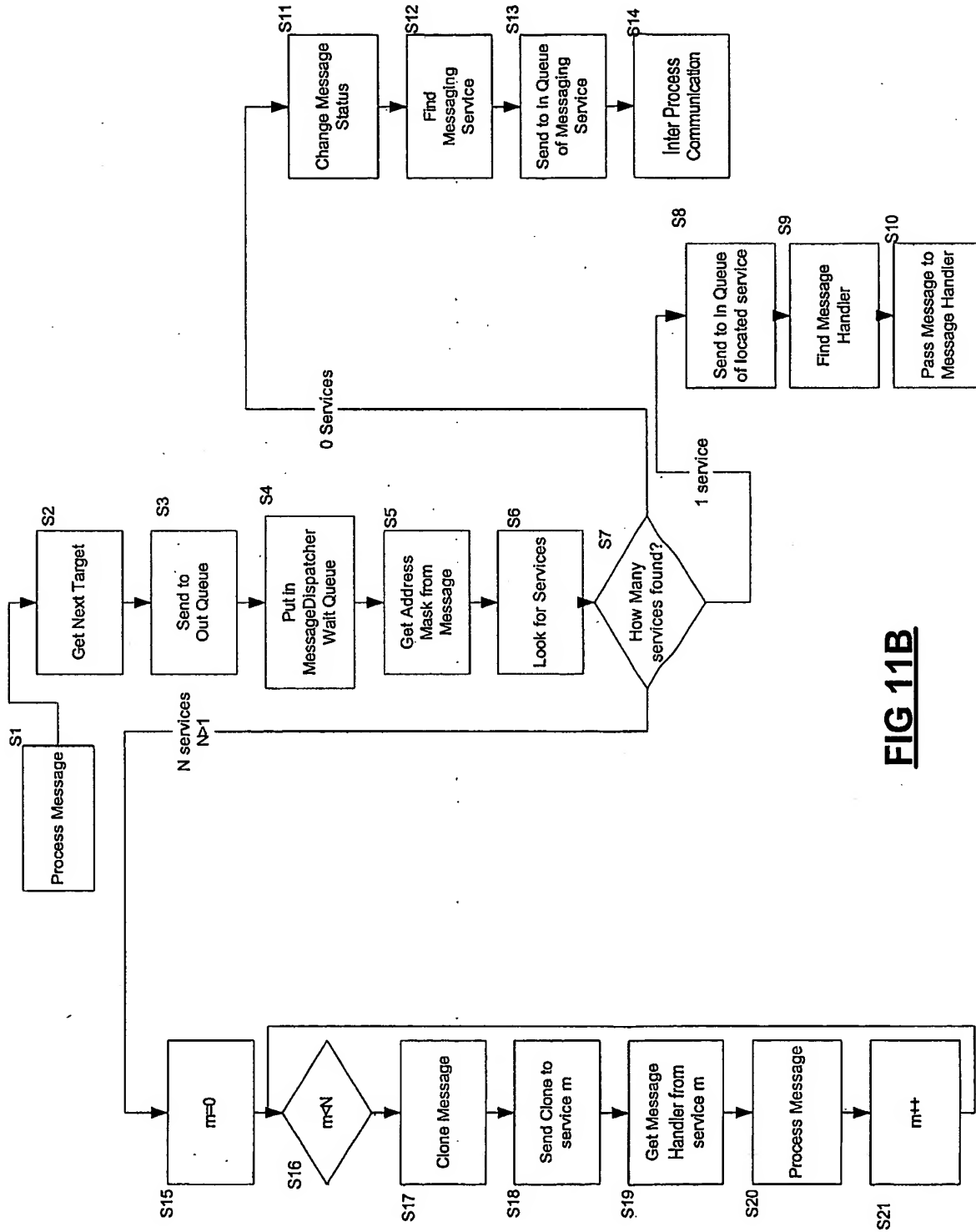
**FIG 10**

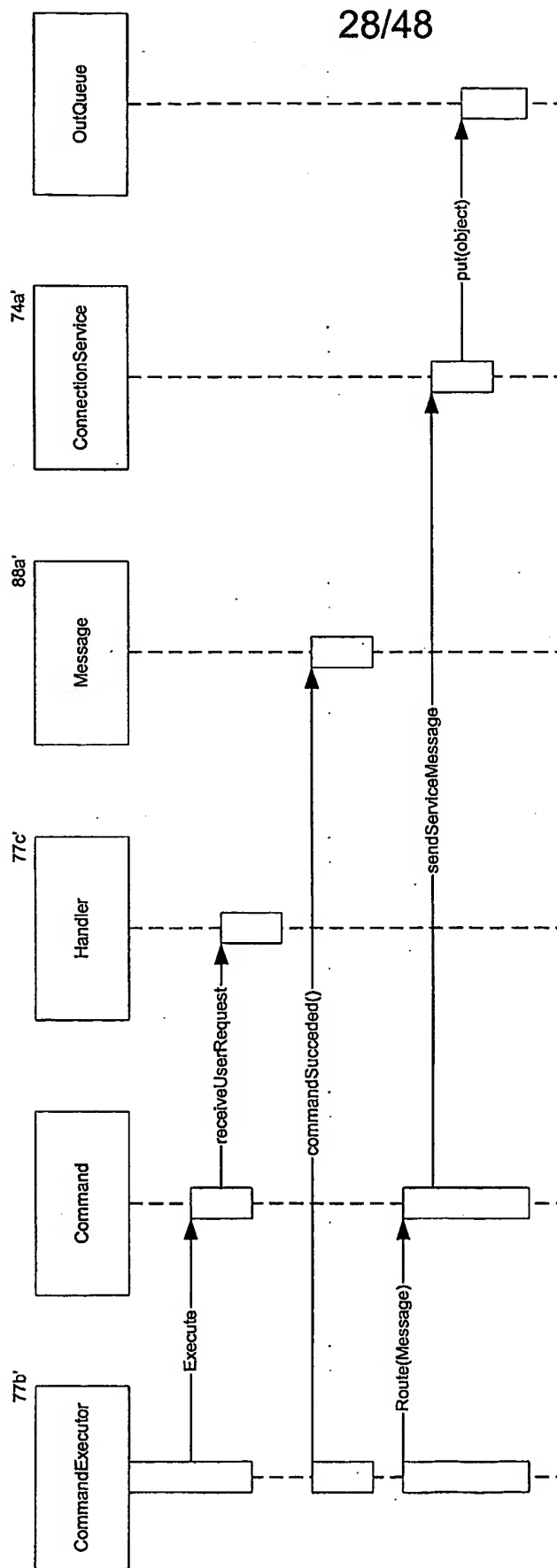
26/48



**FIG 11A**

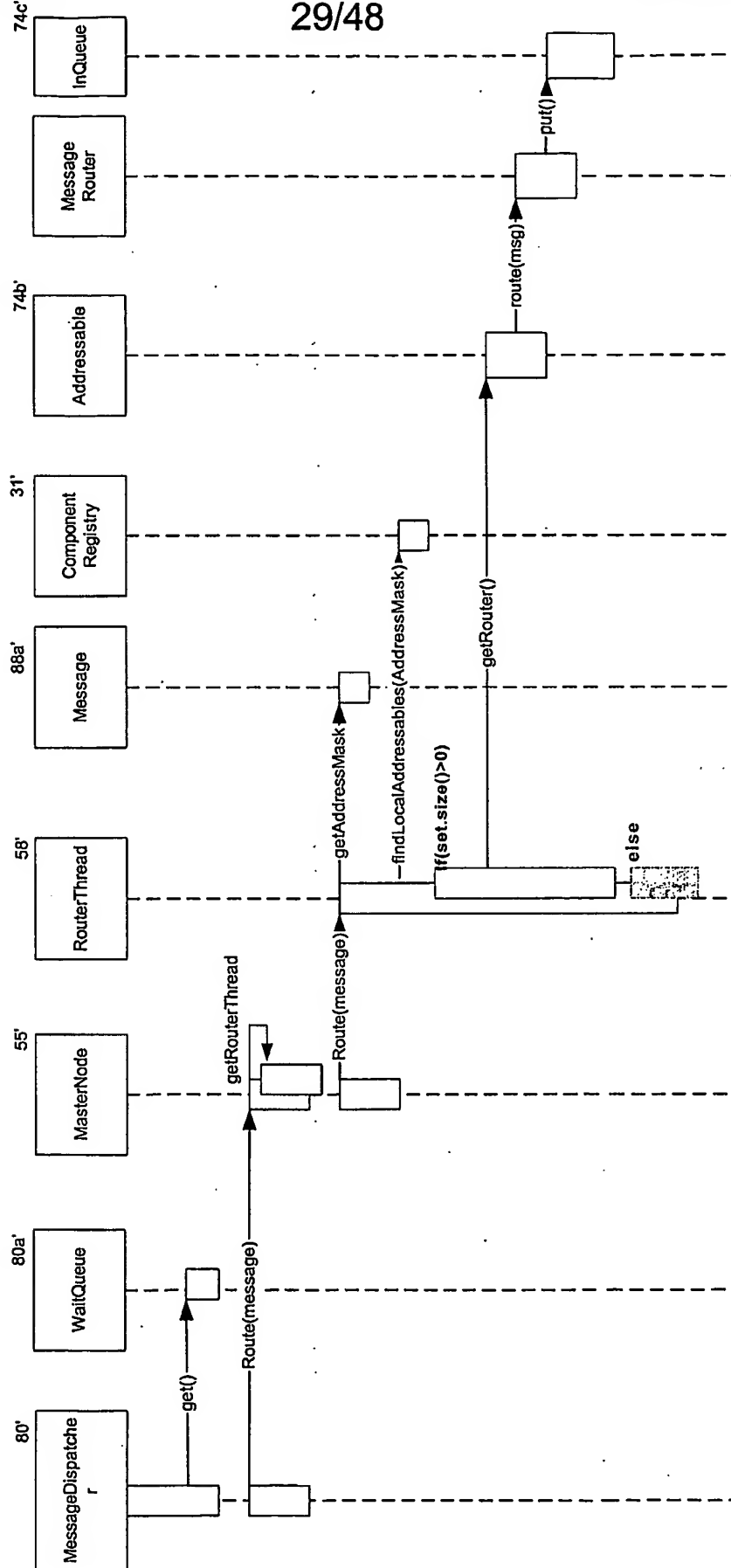
27/48

**FIG 11B**

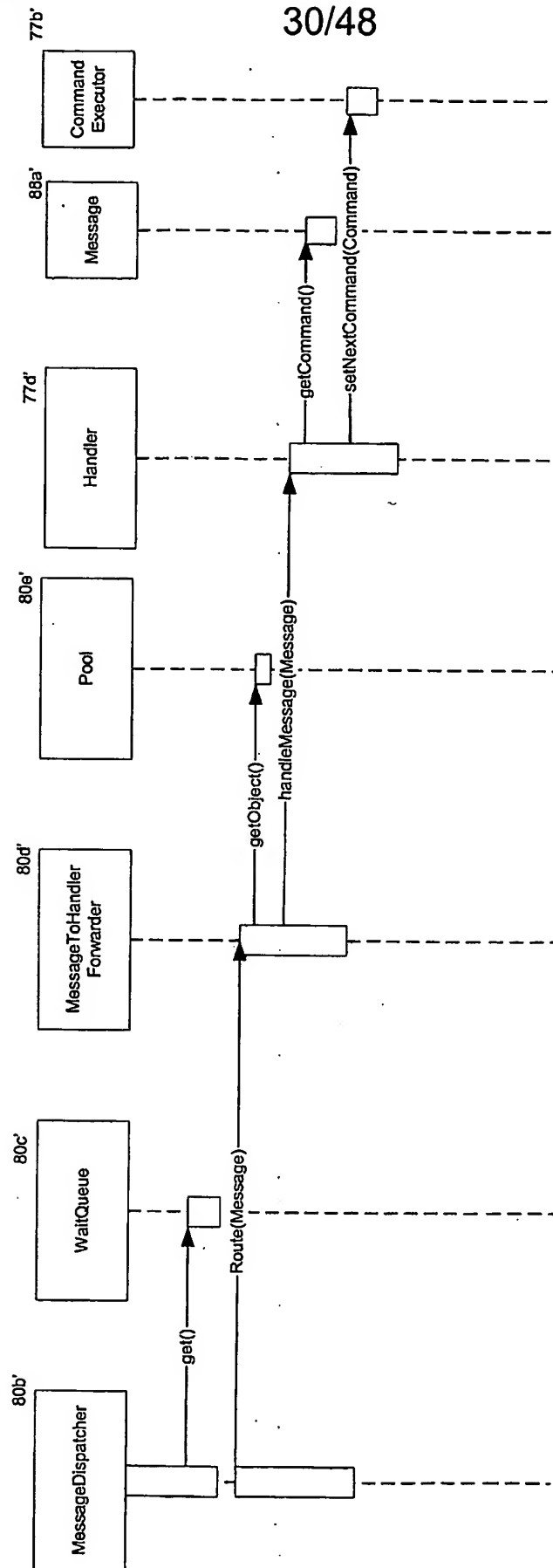


**FIG 12A**

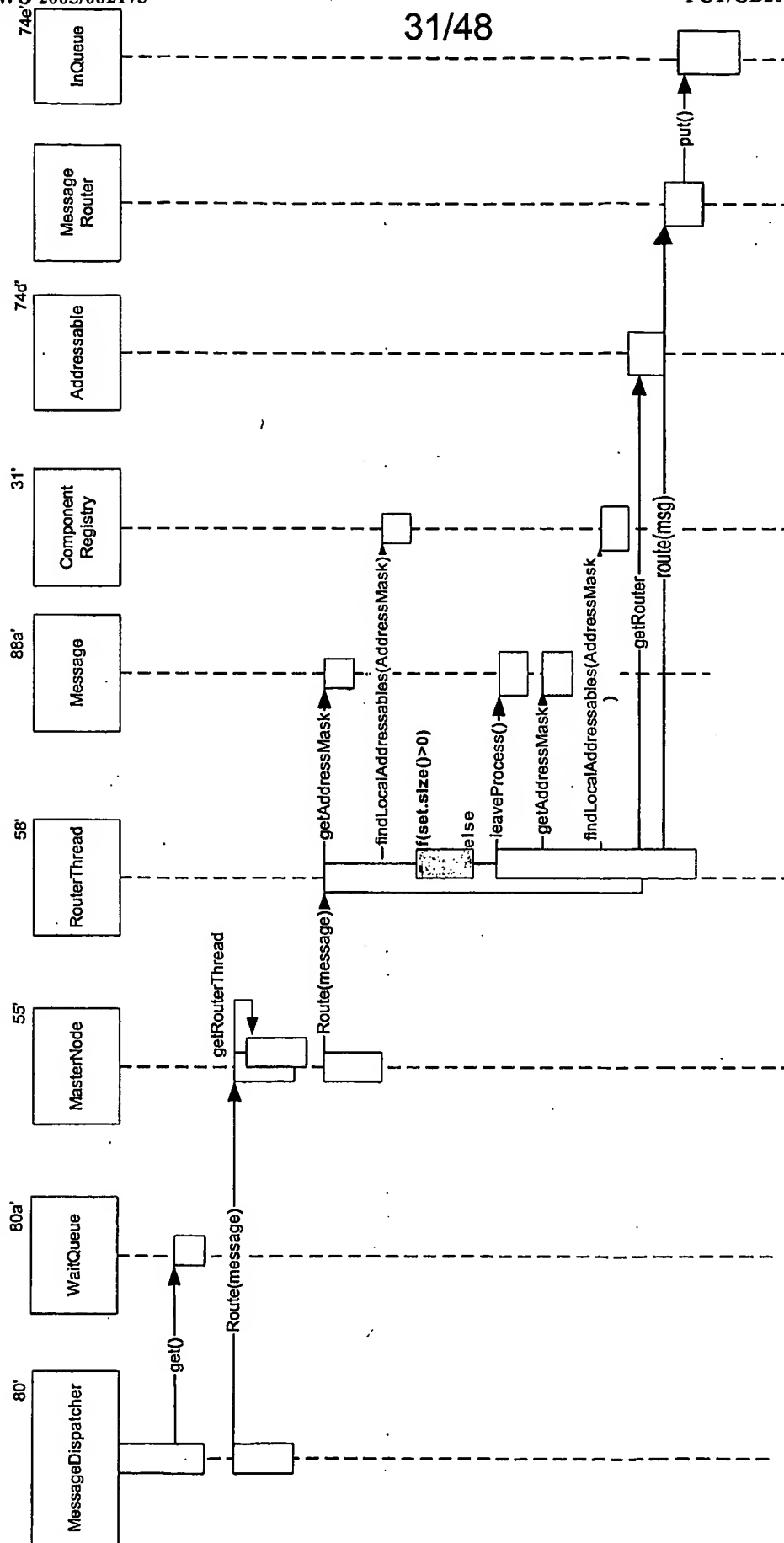
29/48



**FIG 12B**

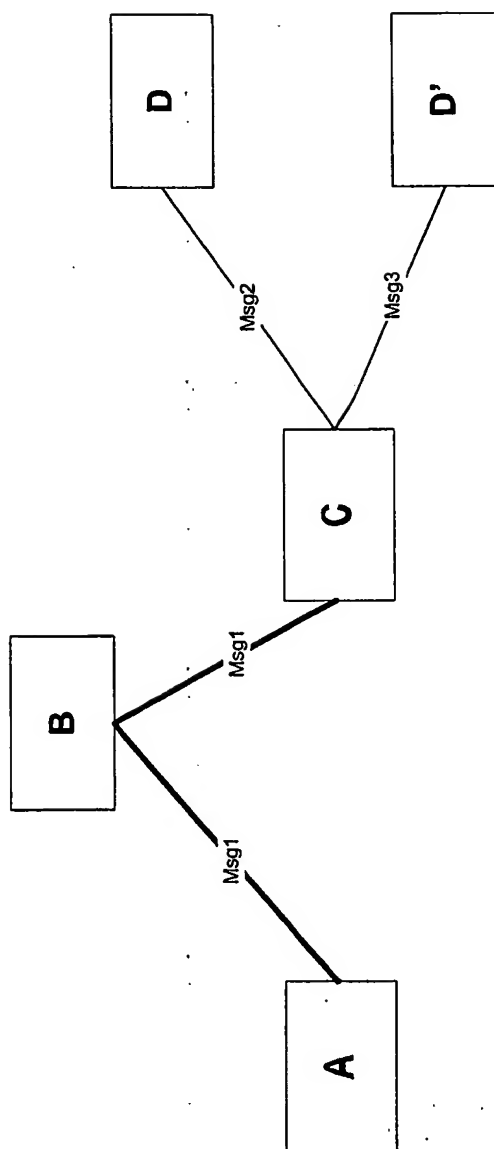


**FIG 12C**



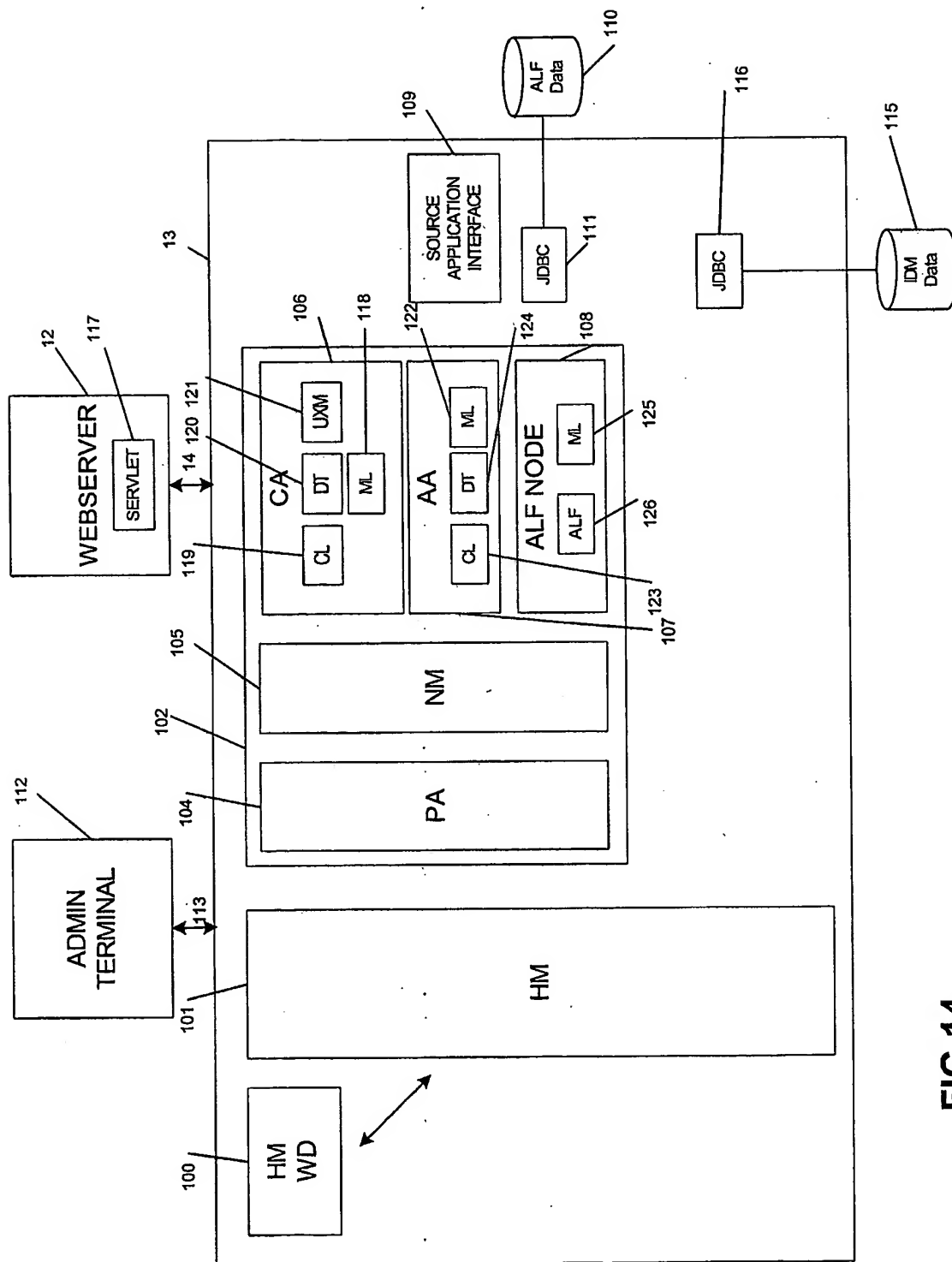
**FIG 12D**

32/48

**FIG 13**

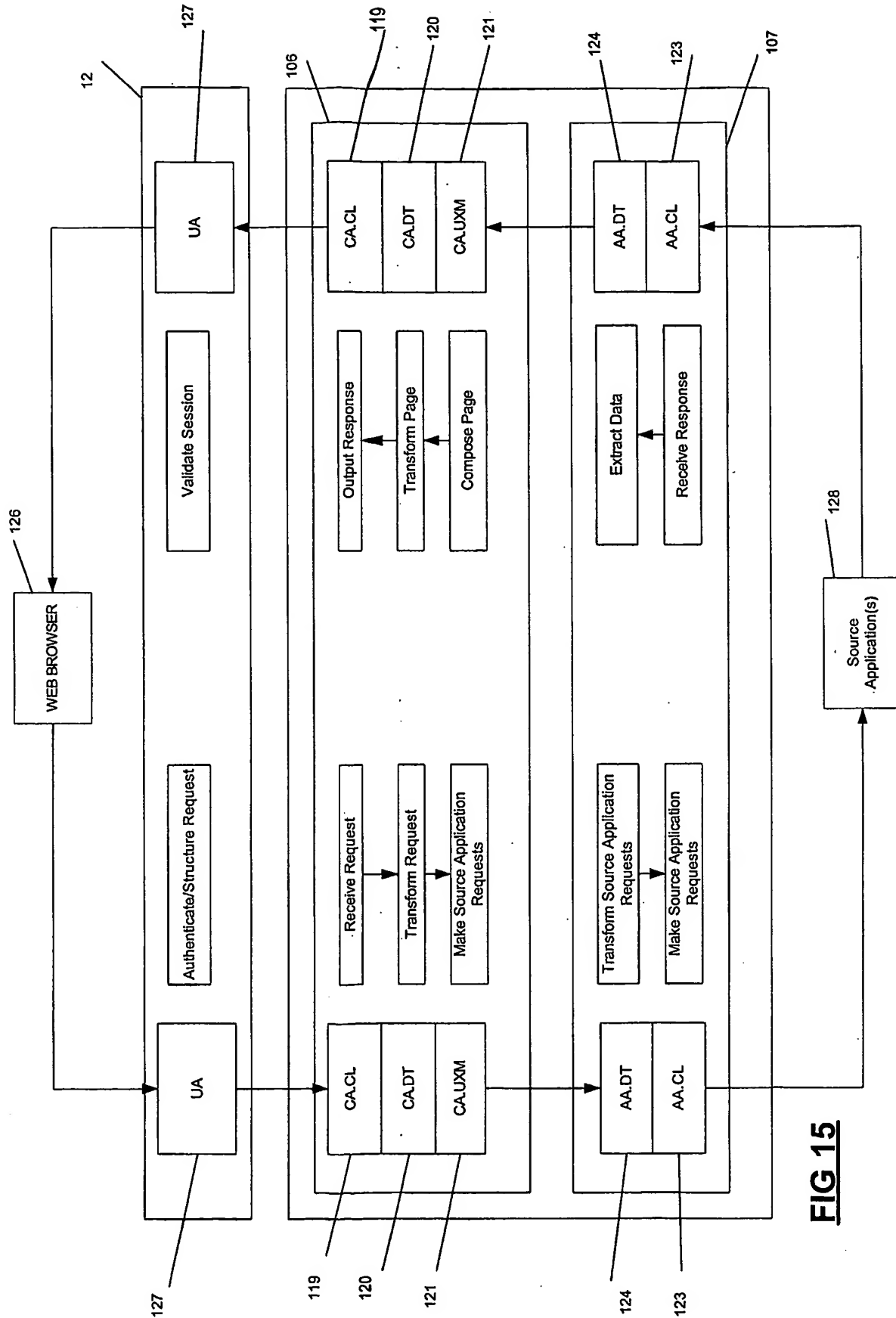


33/48



**FIG 14**

34/48



35/48

Parameter	Req.	Type	Values	Notes
g2jms.config	Y	Str		File containing the jms configuration properties for the Listener e.g. g2jms.config=c:\conf\jms.properties
g2listener	Y	Str		This is the name of the node under the UAconfigs node in the IDM containing the listener configuration. e.g. g2listener=weblistener
g2config.instance	Y	Str		Node in the IDM that has an instance definition for the Composite Application. [shouldn't be necessary but is in current release e.g. g2config.instance=googleweb
g2config.home	Y	Str		Directory containing the IDM properties file for the Composer/Listener. This has to point to a directory containing the file rather than the file. e.g. g2config.home=c:\GCP2.3.0\conf
g2tracer.conf	Y	Str		File containing the logger configuration for the Listener. e.g. g2tracer.conf=c:\conf\listenerlog.conf

**FIG 16**

Parameter	Req.	Type	Values	Notes
G2config.home	Y	Str		Directory containing the IDM properties file for the Composer/Listener. This has to point to a directory containing the 'IDM.properties' file rather than the file itself e.g. g2config.home=c:\GCP2.3.0\conf
g2config.instance	Y	Str		Node in the IDM that has an instance definition for the Composite Application. e.g. g2config.instance=googleweb
g2config.webhost	Y	Str		Host-name (or IP address) and port, of the Web Server used with the Listener. Used to rewrite the pages served up through the composer to point back to the listener. e.g. g2config.webhost=192.168.1.229:80
g2config.baseipport	Y	Str		Specify the Admin console port e.g. g2config.baseipport=8080
g2jms.config	Y	Str		File containing the jms configuration properties for the Composer e.g. g2jms.config=c:\conf\jms.properties
g2basetracer.conf	Y	Str		File name containing the logger configuration(s) for the Listener – It will look for a file per process so there needs to be at least a '0' and a '1' log file. e.g. g2tracer.conf=c:\conf\composerlog.conf

**FIG 17**

36/48

Name	Req.	Type	Values	Notes
APPLICATION_CLASS	Y	Str		Destination service for the incoming user request message that will be generated. This is the application class name used for the composite application.
UXM_MODIFIED	Y	Str	true/false	Set to true if the request is to be processed by the Composer. If false, then the UXM will not process the request and pass it on to the AA (pass through applications)
UXM_TREEID	Y	Str		The 'id' (from the control data-group) of the TreeRootNode targeted by this request.
UXM_NODEID	Y	Str		The 'id' (from the control data-group) of the TreeNode (within the hierarchy defined by UXM_TREEID) targeted by this request.

**FIG 18**

```

hm.g2config.baseport=9080
hm.rmiport=1299
hm.g2config.webhost=10.230.226.162
hm.g2config.instance=MyInstance
hm.g2jms.config=/usr/genient/composer/conf/jms.properties
hm.g2tracer.conf=/usr/genient/composer/conf/composer-log.properties
pa.x.Xms512m
pa.x.Xmx512m
pa.x.XXl:NewSize=64m
pa.x.XXl:MaxNewSize=64m
pa.x.XXl:+DisableExplicitGC
pa.x.g2config.webhost=10.230.226.162
pa.x.g2config.home=/usr/genient/composer/conf
pa.x.g2config.instance=MyInstance
pa.x.g2jms.config=/usr/genient/composer/conf/jms.properties
pa.x.g2tracer.conf=/usr/genient/composer/conf/composer-log.properties
pa.x.java.net.ssl.trustStore=/usr/genient/j2re1_3_1_08/lib/security/g2cert
pa.x.java.net.ssl.keyStoreType=pkcs12
pa.x.java.net.ssl.keyStorePassword=thglnwtfos
pa.x.java.net.ssl.keyStore=/usr/genient/j2re1_3_1_08/lib/security/xF8FF_2_XEG000004.p12
pa.x.java.protocol.handler.pkgs=HTTPClient

```

**FIG 19**

37/48

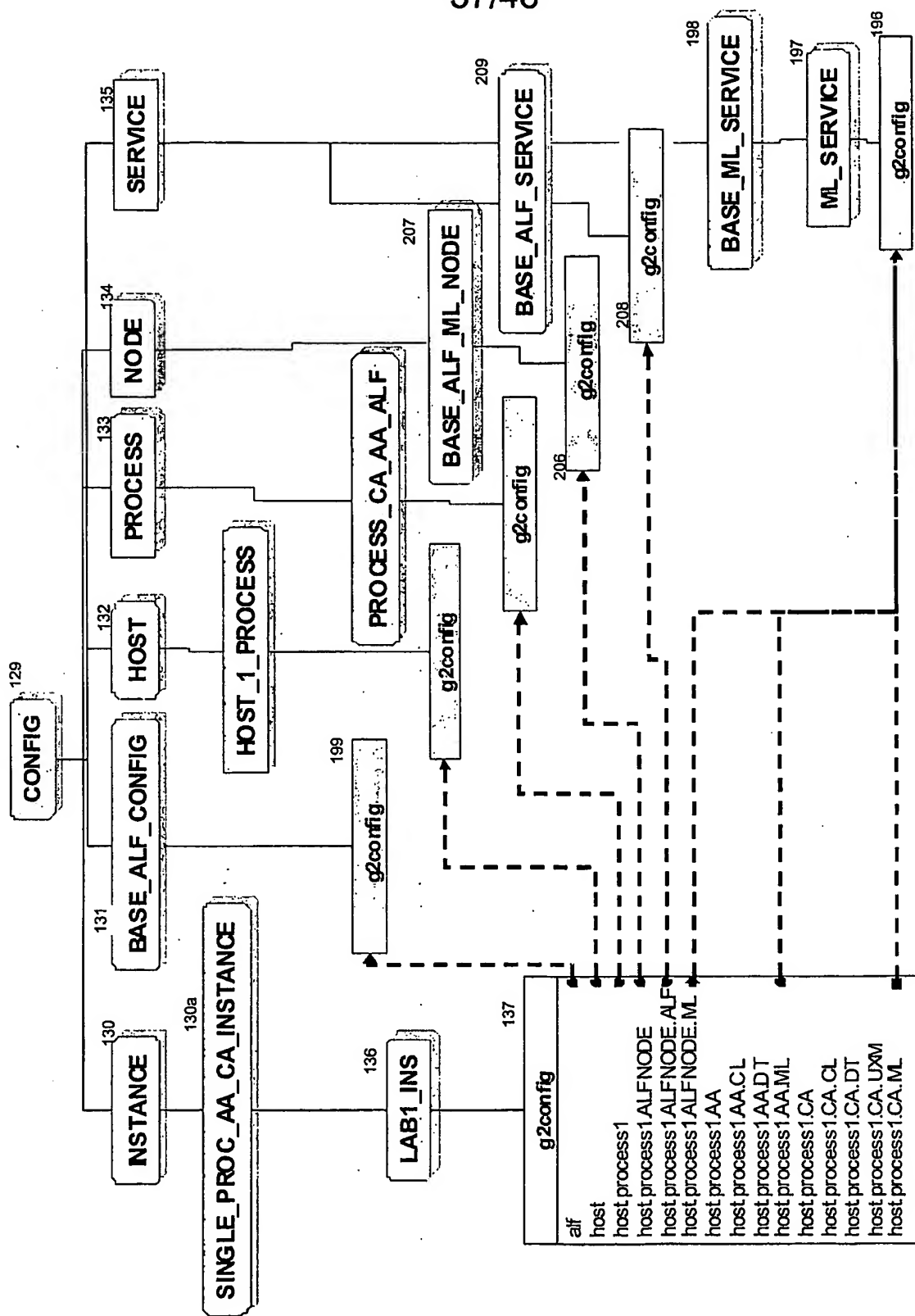
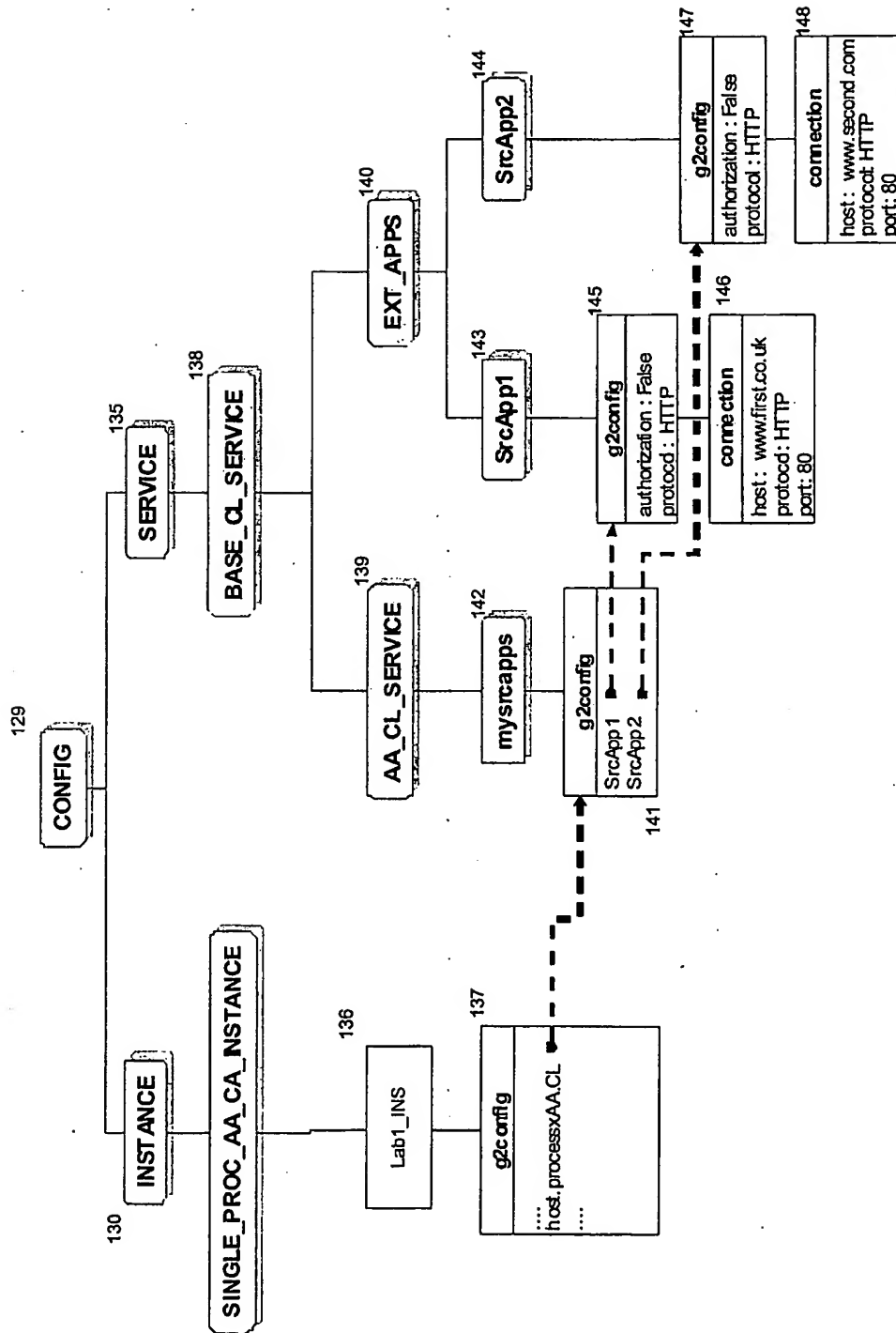


FIG 20

38/48



**FIG 21**

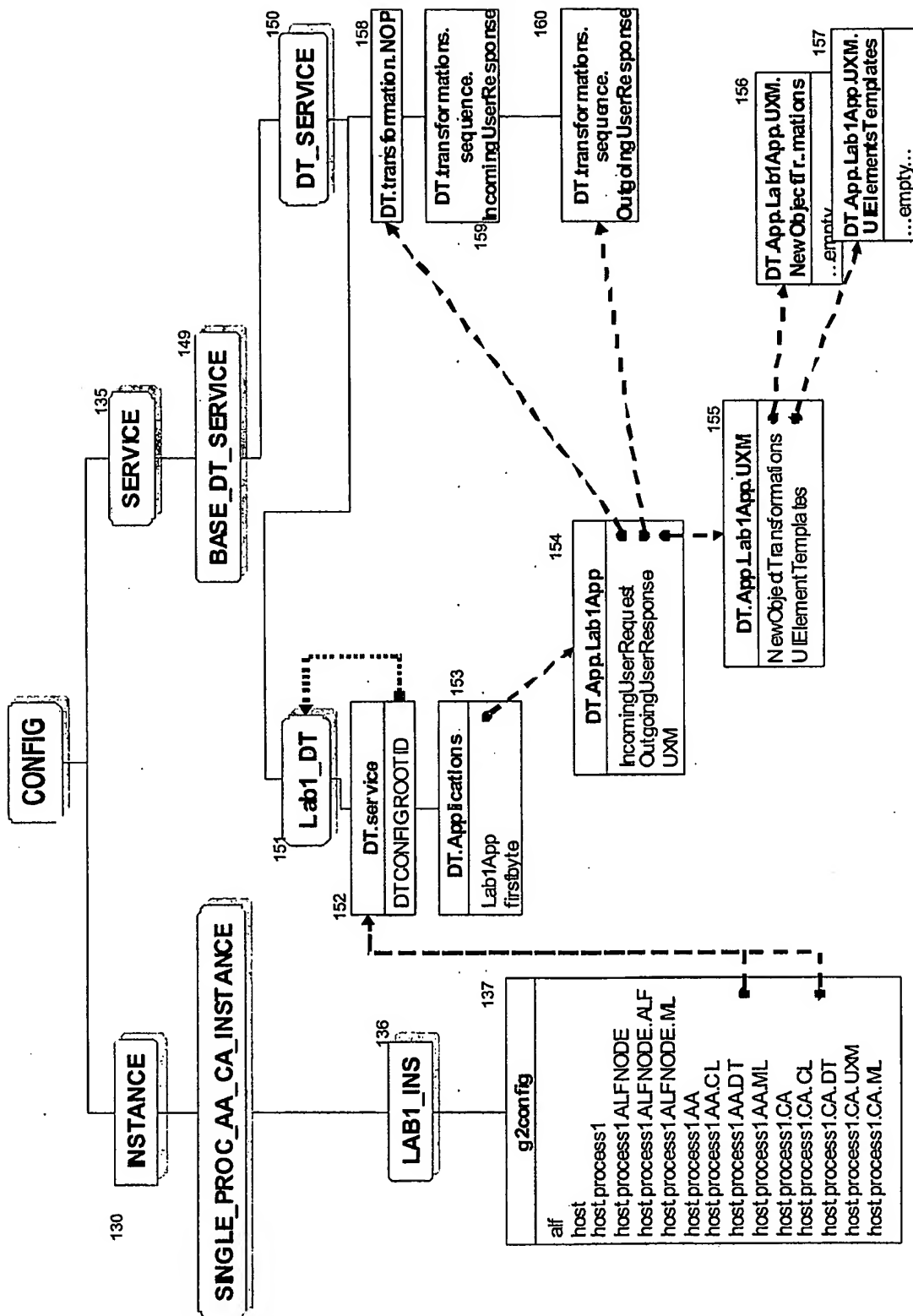
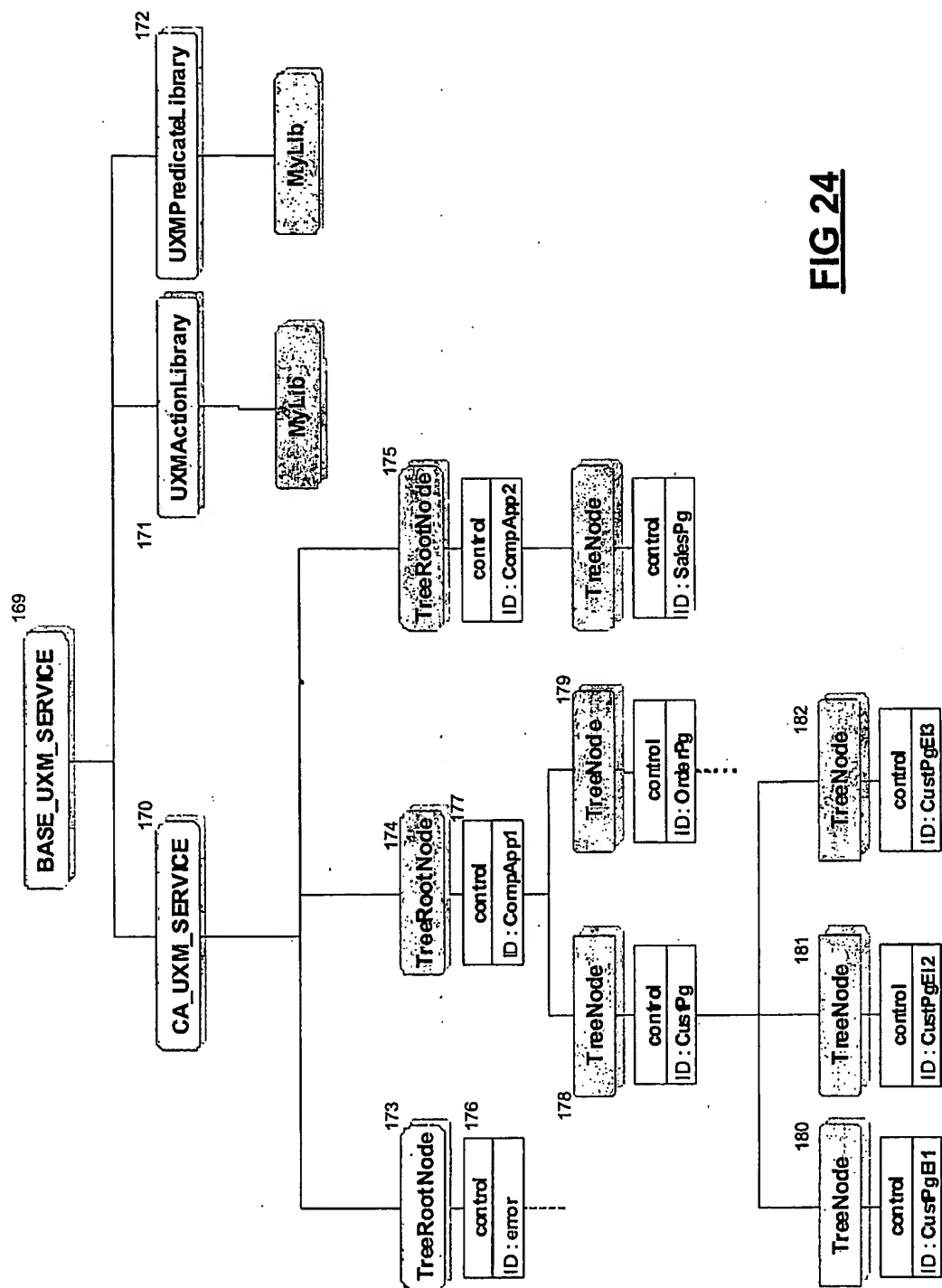


FIG 22



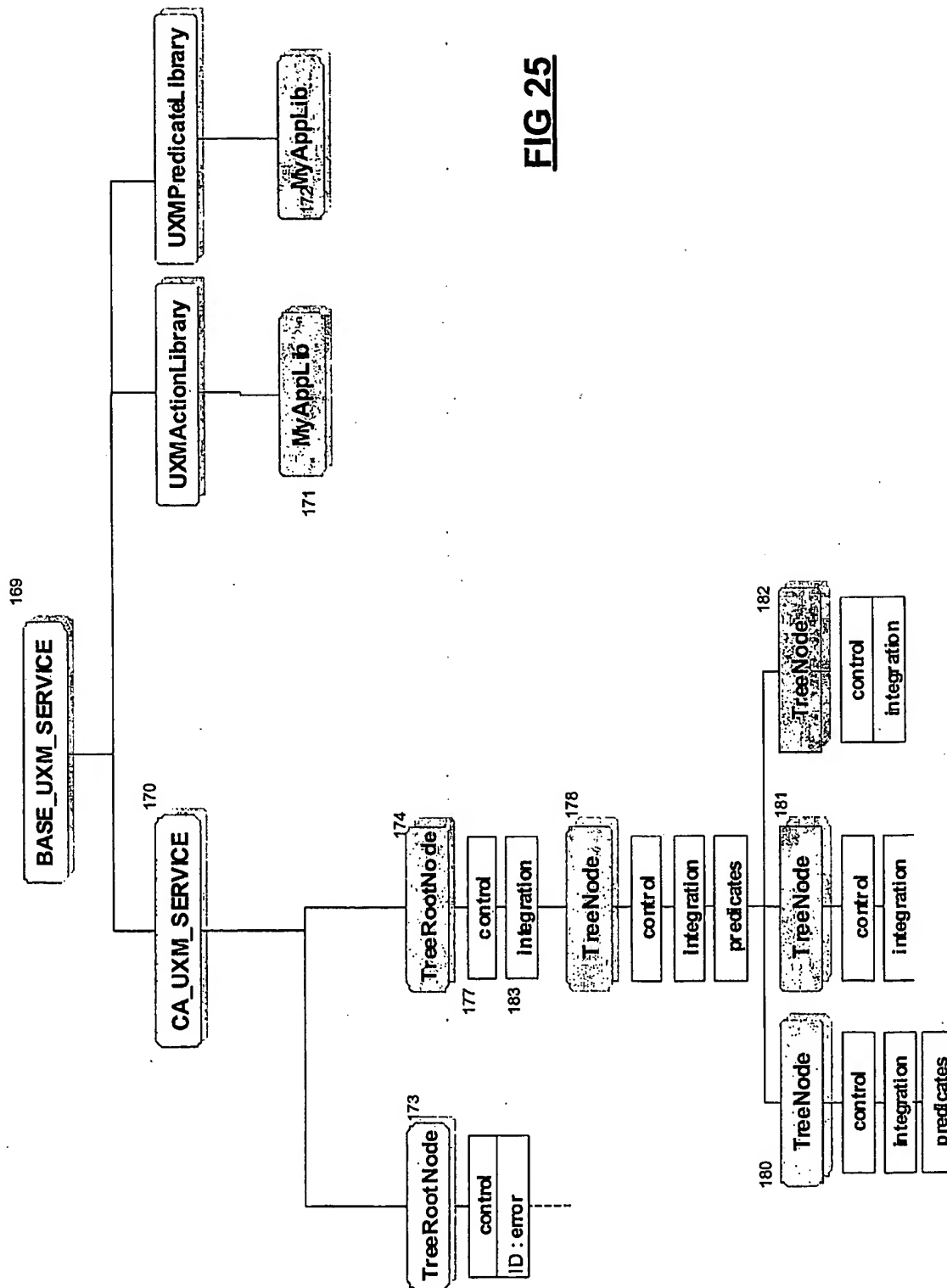


41/48

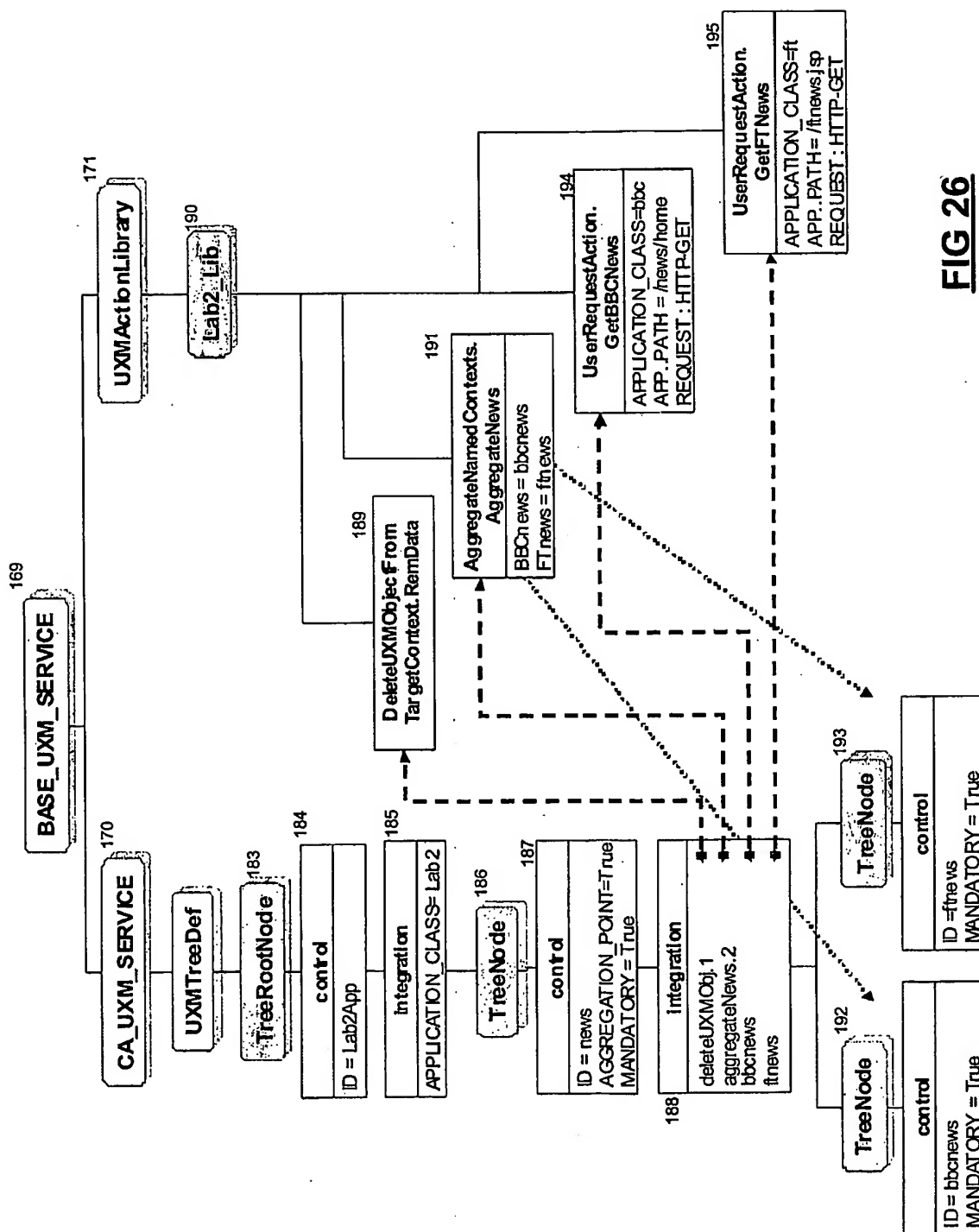


**FIG 24**

42/48

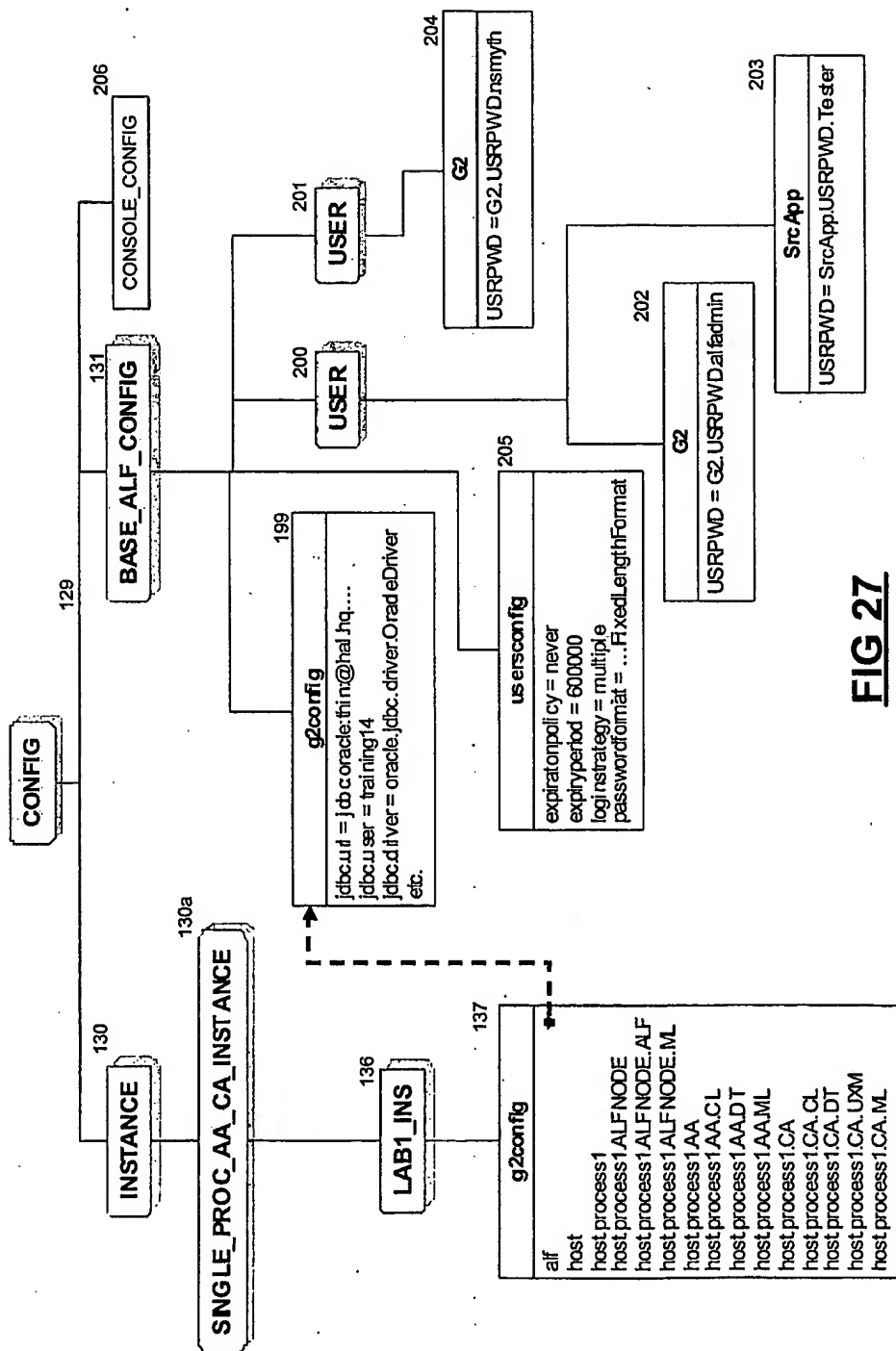


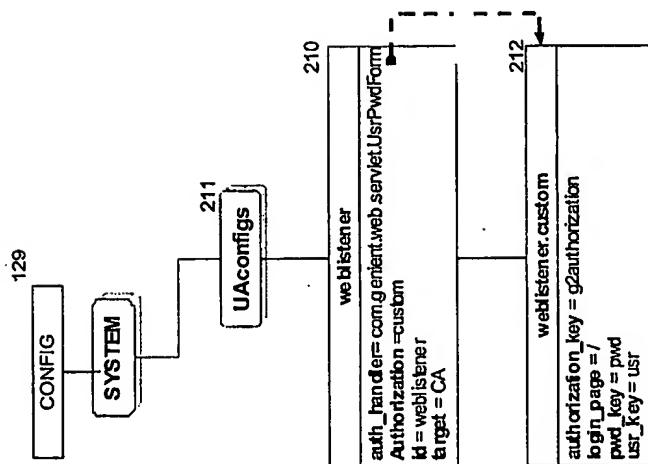
**FIG 25**



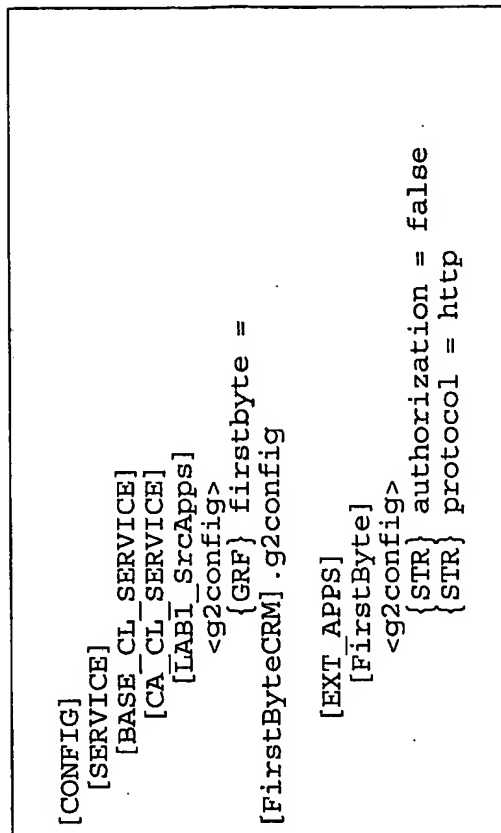
**FIG 26**

44/48



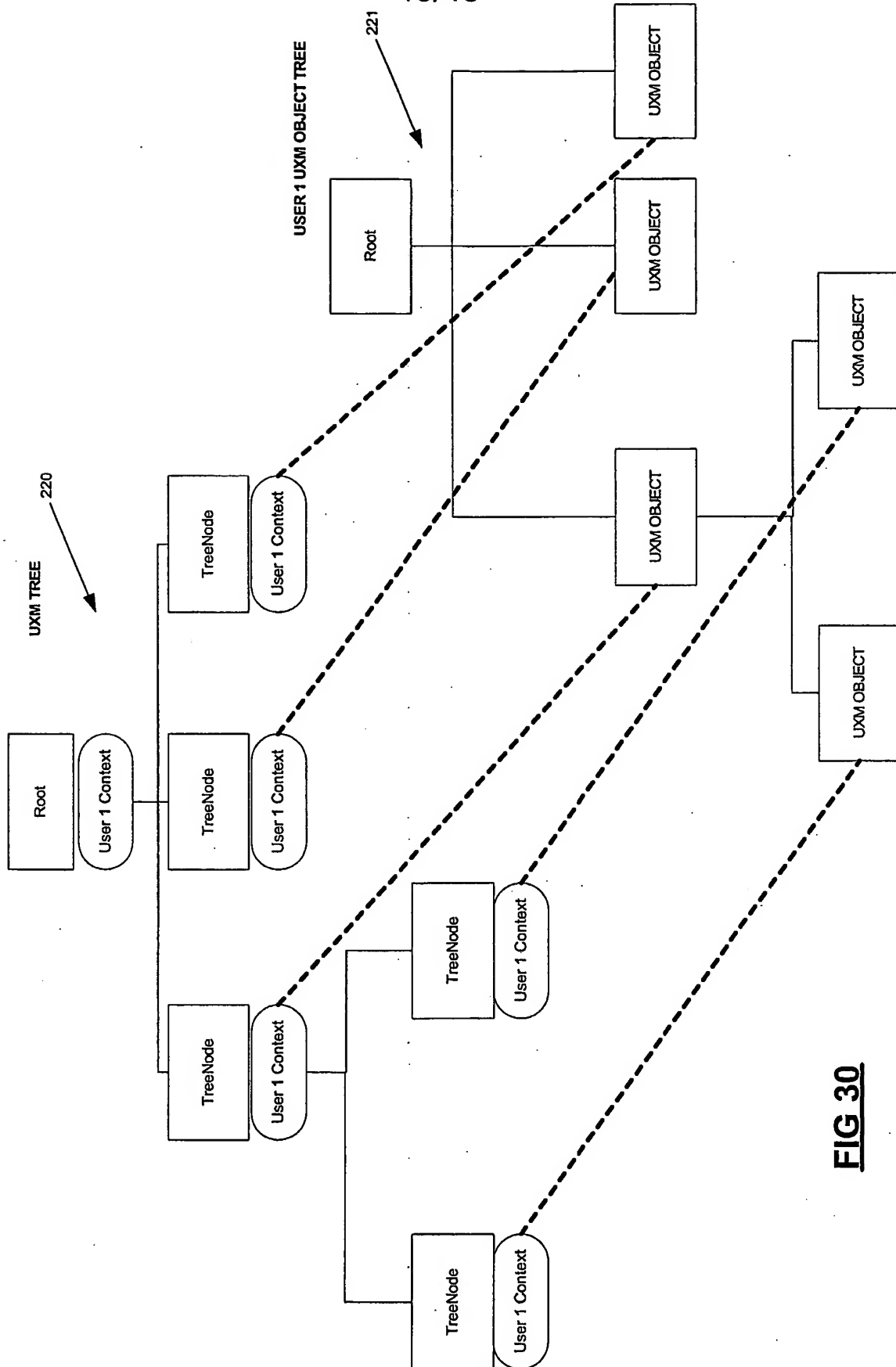


**FIG 28**

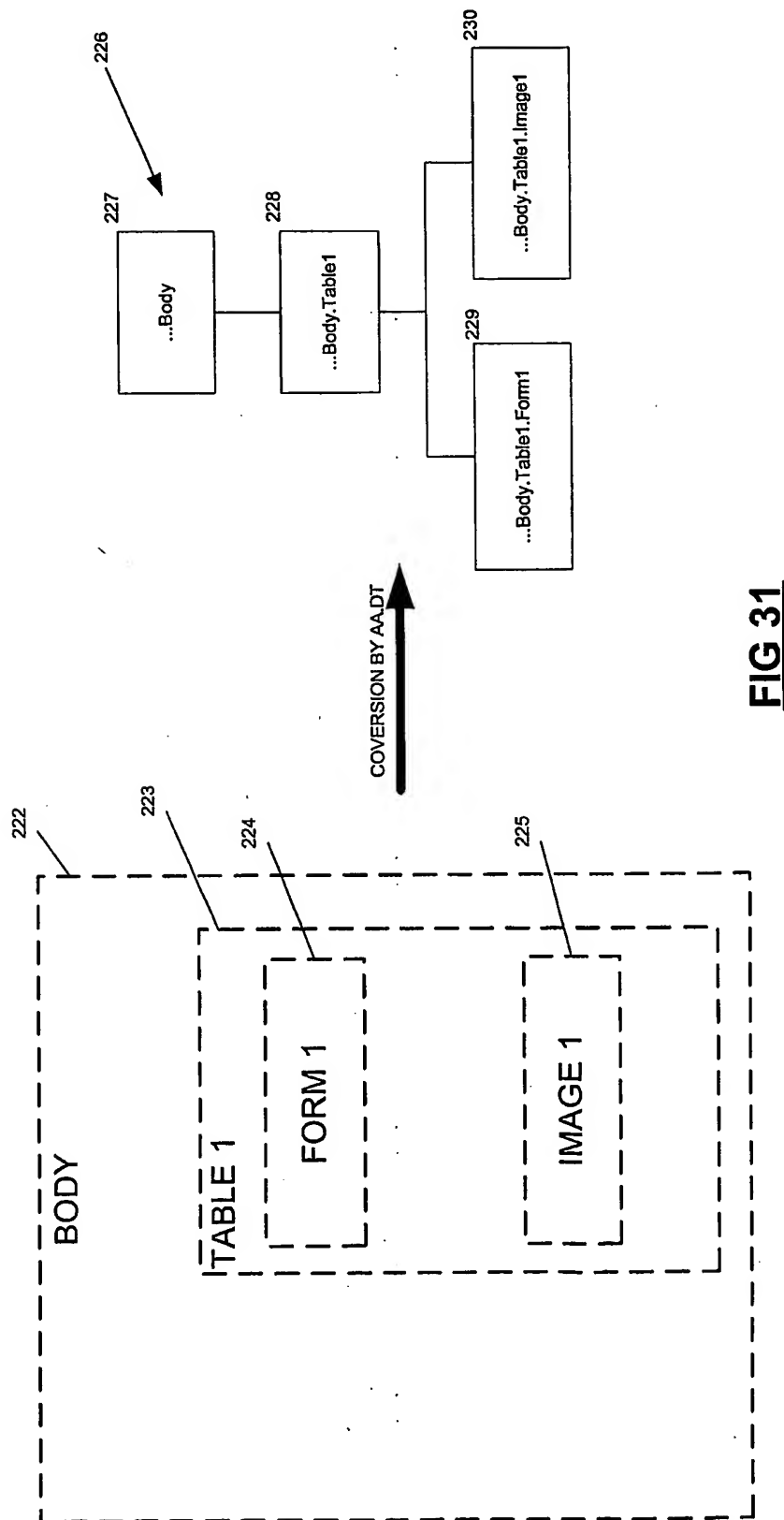


**FIG 29**

46/48

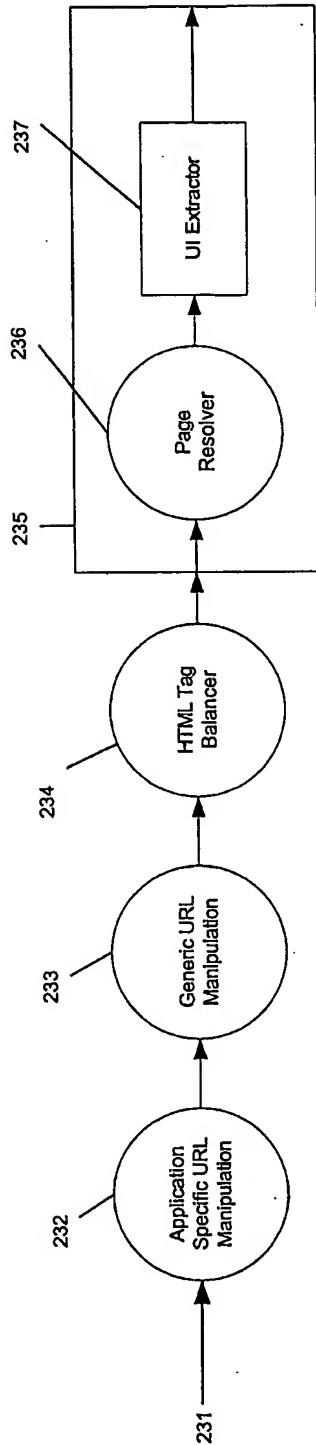


**FIG 30**

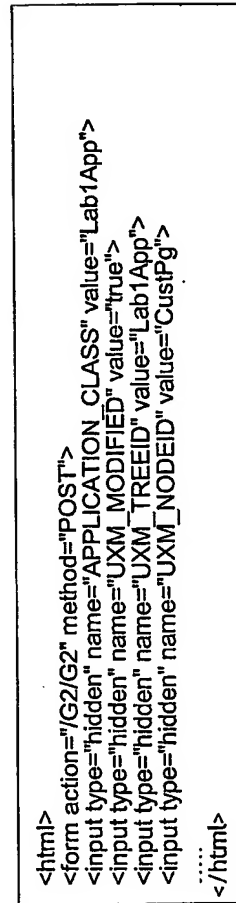


**FIG 31**

48/48



**FIG 32**



**FIG 33**